

PARDUS İLERİ SEVİYE EĞİTİMİ

Eylül 2012

Eğitim ve Materyal Hakkında

- Bu eğitim materyali 3-14 Eylül 2012 tarihlerinde MEB Öğretmen Yetiştirme ve Geliştirme Genel Müdürlüğü tarafından Mersin'de düzenlenen 369 Numaralı «Pardus İleri Seviye» kursu için hazırlanmıştır.
- Kurs katılımcılarının giriş ve orta düzey sistem yönetim kabiliyetleri kazanmaları hedeflenmektedir.
- Kursun hedef kitlesi ve amaçları genel amaçlı bir sistem yönetim kursu ile ortak noktalar içermekle birlikte farklılıkları da bulunmaktadır. Bu nedenle ders planı, içeriğin oluşturulması, sunum biçimi alternatif kaynaklar ile çeşitli farklar içerebilir.
- Bu materyalde kullanılan temel Linux dağıtımı Pardus 2011.3K RC1 (15 Ağustos 2012) dağıtımıdır. Söz konusu dağıtım Debian temelli bir dağıtım olduğu için bu materyal genel olarak Debian temelli herhangi bir dağıtım için yararlı içeriğe sahiptir.

Eğitimin Öğrenme Hedefleri ve Ana Başlıkları

Öğrenme Hedefleri

- Linux ve özgür yazılım geliştirme sürecini sistem yöneticisi perspektifinden anlamak
- Linux işletim sisteminin temel yapılarını belirli bir detay düzeyinde öğrenmek
- Temel sistem yönetim araçlarını tanımak
- Temel ağ protokollerini anlamak ve bu protokollere dönük sunucuları yapılandırabilmek

Ana Başlıklar

- Bilişim Teknolojileri, Özgür Yazılım ve Linux
- Linux'un Detayları
- Sistem Yönetim Süreci
- Temel Ağ Protokolleri ve Servisleri

Eğitimin Öğrenme Hedefleri ve Ana Başlıkları

Öğrenme Hedefleri

- Sanallaştırma araçlarını tanımak ve etkin kullanabilmek
- Güvenlik duvarları ile temel faaliyetleri gerçekleştirmek
- Özgür yazılım süreçlerine katkı sağlayabilmek için takip edilmesi gereken yöntemleri ve kullanılacak araçları tanımak
- Daha ileri düzeydeki faaliyetlere dönük bir yol haritası oluşturmak

Ana Başlıklar

- Sanallaştırma
- Güvenlik Duvarları
- Özgür Yazılıma Katkı Sağlamak
- Daha Sonrası?

Özetle Ana Başlıklar

- Bilişim Teknolojileri, Özgür Yazılım ve Linux
- Linux'un Detayları
- Sistem Yönetim Süreci
- Temel Ağ Protokolleri ve Servisleri
- Sanallaştırma
- Güvenlik Duvarları
- Özgür Yazılıma Katkı Sağlamak
- Daha Sonrası?

1. Bölüm

Bilişim Teknolojileri, Özgür Yazılım ve Linux

- Bu bölümde, bilişim teknolojilerini «ürün geliştirme» perspektifinden inceleyerek özgür yazılım ve Linux'un bu alandaki etkilerini tartışılacaktır.
- Tartışma kısa bir tartışma olmakla birlikte, sistem yöneticilerinin «ürün yaşam çevrimi yönetimi» görevleri de bulunmakta olduğu için önemli bir konudur.

Bir Ürün Nedir?

- Bir ürün, elle tutulan veya tutulamayan nitelikleri bulunan bir mal, hizmet veya fikir; veya bu üçünden oluşan bir pakettir.
- Ürünler bir maliyet sonucu üretildikleri ve o ürünü kullanmak isteyen kişilere ulaştırılmaları ayrı bir maliyet unsuru olduğu için ticari birer varlıktır.
- Her ürün için bir «edinme maliyeti» bulunur. Bir ürünün kullanıcısı, o ürünün edinme maliyetini
 - Doğrudan parasal olarak ödemek olabilir
 - Başkasına (örneğin işverenine) karşılatmış olabilir
 - Dolaylı yollardan (örneğin ürünün reklamını yaparak) karşılıyor olabilir.
- Ürünlerin üreticileri, dağıtıcıları ve kullanıcıları arasındaki ilişkileri düzenleyen yazılı, sözlü veya doğrudan ifade edilmese bile sosyal olarak uyulan kurallardan oluşan bir sözleşme bulunmaktadır.
- Ticari kuruluşlar bir yazılı sözleşme bulunması taraftarıdır. Ancak ticari başarıları açısından yazılı olmayan sözleşmeleri de tercih edebilir veya kullanabilirler.

Bir Ürün Nedir?

- Ürünlerin bir ihtiyacı karşılaması esastır. Bu ihtiyaç zaman zaman çok net biçimde ifade edilmiş olabileceği gibi zaman zaman kolayca ifade edilemeyebilir veya ihtiyaç sahibi bu ihtiyacını dile getirmekte zorluk çekebilir.
- Ürünün müşteriler tarafından tatmin edici bulunması, ürünün hitap ettiği ihtiyacı ne derece karşıladığı ile ilintilidir. Ancak aynı ürün, benzer olsa da belli düzeyde farklı ihtiyaçları olan çok sayıda müşterinin ihtiyaçlarını aynı derecede karşılayamaz. Bu nedenle tek tek her müşterinin ürünler ile olan ilişkisi farklı derecede tatmin yaratır.
- Müşteri ihtiyaçlarını tespit edip o ihtiyacı doğru biçimde karşılamakta yetersiz kalan ürünler genellikle müşteriler tarafından tatmin edici olarak karşılanmaz.

Ürün Yaşam Çevrimi

- Bir ürünün yaşam çevrimi iki şekilde tanımlanabilir.
 - Seri üretime konu, maddi varlığı bulunan ürünlerde, ürünün tek bir kopyasının müşteri tarafından alınmasından başlayarak o müşterinin ürünü kullanma döneminin tamamını kapsayan süre içinde müşterinin ürün ile olan etkileşimi ürün yaşam çevrimidir.
 - Daha geniş bir tanım, ürüne olan ihtiyacın tespit edilmesinden başlayarak, ürünün tanımlanması, araştırma, tasarım, prototipleme, üretim planlama ve üretim aşamalarından geçmesi, dağıtımı, paketlenerek sunulması, kullanımı gibi geniş bir zaman diliminde yer alan tüm olaylar ve faaliyetleri ürün yaşam çevrimi olarak görür.
- Yazılım ve yazılım temelli ürünlerin yaşam çevrimlerinden bahsederken daha çok ikinci tanım kullanılır.

Ürün Yaşam Çevrimi

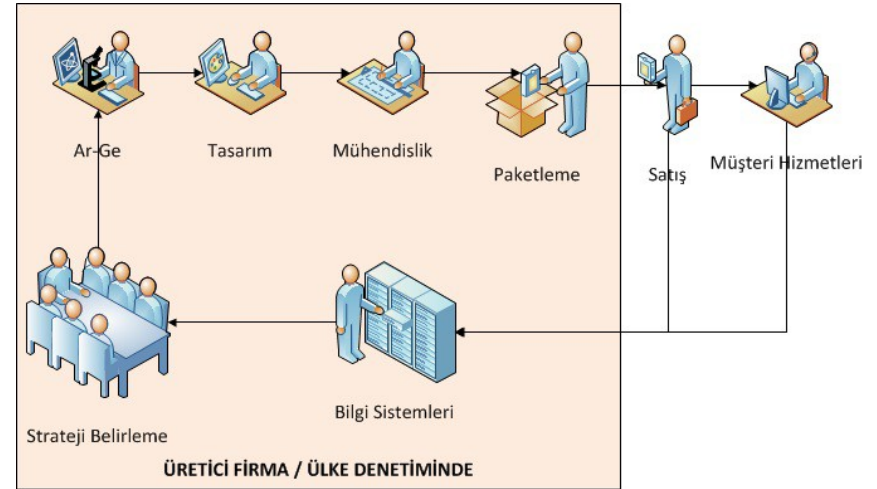
- Yazılım ürünleri için yaşam çevriminin geniş biçimde tanımlanması, müşterilerin tatmin edilmesi için de önemlidir.
 - Yazılım ürünlerinin bir bölümü belirli bir müşteri için «proje temelli» olarak üretilen ürünlerdir. Bu ürünlere «proje teslimatı» gözüyle bakmak gerekir. Dolayısı ile üretici, proje süresince müşterinin ihtiyaçlarının ne şekilde değiştiğini izlemeli ve gerekli görürse teknik olanaklar dahilinde üründe değişikliklere gitmelidir.
 - Yazılım ürünlerinin bir diğer bölümü ise sabit bir dizi özelliğe sahip olan ürünler olarak pazara sunulan ve çok sayıda müşteriden farklı geri bildirim alan, bu müşterilerin önemli bir çoğunluğunu tatmin etmeyi hedefleyecek şekilde üretilen ürünlerdir.
 - Zaman zaman bazı yazılım ürünleri bir kategoride üretilip bir diğer kategoride kullanılabilir. Bu durumda üretim ve tüketimin biçimleri arasındaki uyumsuzluk sorunlar yaratacaktır.

Ürün Yaşam Çevrimi

- Ayrıca üründen elde edilen tatmine ek olarak «ürünün edinilme süreci» boyunca elde edilen tatmin de çok önemlidir. Ürün yaşam çevrimi bu konuyu da ele almalıdır.
 - Örnek olarak «ihale» ile edinilmesi gereken bir ürün için üretici ihaleye iştirak edemiyor ise, müşterilerini tatmin etmesi baştan mümkün olmayacaktır. Bu durumda üreticinin ve ürünün «ihaleye katılma» amaçlı olarak değişmesi ve gelişmesi gerekir.
 - Üretici, bu amaçla ürününde değişikliklere gidebilir . Bu değişiklikler aslında son kullanıcıyı daha az tatmin edecek bir ürüne neden olsa dahi, üretici bunu yapmak zorunda kalacaktır.
- Prensip olarak ürün yaşam çevrimine müşterinin veya kullanıcının ne aşamada dahil olduğu önemlidir.

Ürün Yaşam Çevrimi

- Prensipte olarak ürün yaşam çevrimine müşterinin veya kullanıcının ne aşamada dahil olduğu önemlidir.
 - Klasik üretim süreci ve ürün yaşam çevrimi, üreticinin çevresini denetim altında tutmaya çalışmasına dayalıdır ve üreticinin faaliyetleri tarafından güdülür.
 - Bu durumda başarılı üreticiler değer zincirine neredeyse tamamen hakim olurlar.
 - Müşteri bu türdeki bir süreçte hangi aşamada dahil olursa olsun, ürün konusundaki son sözü daima «üretici» söyler.



Ürün Yaşam Çevrimi

- Ürün yaşam çevriminin üretici güdümlü olmasına alternatifler de vardır. Ürün güdümlü veya müşteri güdümlü yaşam çevrimleri de mümkündür.
- Her bir yaşam çevrimi süreci farklı alanlara odaklanır ve aslında hiç bir süreç tek başına ideal süreç değildir. Önemli olan üreticinin müşterisi ile olan ilişkisini başarı ile sürdürmesi için ne türde bir ilişki kurmasının ve nasıl bir ürün yaşam çevrimi takip etmesinin doğru olacağıdır.
- Bu nedenle bir kuruluşun bir ürüne olan bakış açısı ve o ürünle ilgili eylemleri o kuruluşun kendi perspektifine bağlıdır.

Teknoloji Nedir?

- Teknoloji araçların, tekniklerin, yöntemlerin, sistemlerin, örgütlenme biçimlerinin, var olan sorunlara yaklaşımların ve çözümlerin üretilmesi, değiştirilmesi, kullanımı ve bu eylemlerin bilgi olarak oluşturulması sürecidir.
 - Örneğin bir kullanıcının bir diğer ile eşzamansız mesajlaşması sorunu insanlık kadar eski bir sorundur. Geleneksel posta alt yapısı bu soruna bir çözümdür. Bu çözümün tasarımı ve uygulanması sırasında çok sayıda teknik (örneğin zarfların damgalanması), yöntem (posta kodları kullanarak zarfların önce dar bölgelere sevk edilmesi), örgütlenme yapısı (postaneler) oluşturulmuştur.
 - Daha sonra elektronik postanın geliştirilmesi sırasında bu bilgiler ve deneyimler kullanılmıştır. Elektronik postalar bir «zarf» içermese dahi, zarfta en başta yazan bilgiler (gönderen, alıcı, gönderim tarihi, vb) ile zarfın üzerine basılan damgalar (geçtiği postanelerdeki işlem bilgileri) elektronik ortamdaki denklemleri (yazılımların bıraktığı kayıtlar) ile yer değiştirmiştir.

Teknoloji Nedir?

- İnsanlığın ilk teknoloji kullanımı keskin uçlu taşların alet olarak kullanımı ve ateşin ısınma amaçlı kullanımınıdır.
- Teknolojiler, var olan sorunlara yeni sorunlar eklendikçe veya bir sorun yapı değiştirdikçe gelişmektedir. Aynı zamanda bir işi yapmanın daha iyi yollarını bulma çabası da yeni teknolojilerin gelişmesine neden olur.
- Teknoloji gelişimindeki hızın artması aslında karşılaşılan problemlerin de arttığını göstermektedir.
- Teknoloji kullanımındaki en önemli devrim ise araç üretmek için teknoloji kullanmaktır.
 - Bir teknolojinin gücü, çoğu kez o teknolojinin ihtiyaç duyacağı araçların üretilmesinin ne kadar kolay, hızlı veya ucuz olması ile ilintilidir.

Teknoloji Nedir?

- Teknolojinin üretilmesi ve kullanımı bilim ve mühendislik ile yakından ilgilidir.
- Ancak bir teknolojinin farklı kullanım alanları bulması yoluyla çoğaltılması genellikle ilk önce teknolojinin son kullanıcıları tarafından deneyimlenir.
 - Bu deneyimin, bilimsel yöntem ile incelenmesi sonucu, bu teknolojinin söz konusu farklı kullanım alanında güvenli ve verimli bir biçimde kullanımı için gerekenler bilim insanları ve mühendisler tarafından tespit edilir ve sağlanır. Teknolojinin yeni alandaki kullanımını yaygınlaştıran ve ekonomik değer ortaya çıkartan da güvenli ve verimli kullanım olanağının sağlanmasıdır.
 - Dolayısı ile bilim ve mühendislik alanındaki eksikler teknoloji alanındaki yeniliklerin ilk olarak ortaya çıkmasına engel olmamakla birlikte, o yeniliğin yaygınlaşmasını sağlayan ve bu yaygınlaştırmadan ekonomik sonuç elde edecek olan kişinin bilim ve mühendislik açısından yeterli olması gereklidir.

Teknoloji Yaşam Çevrimi

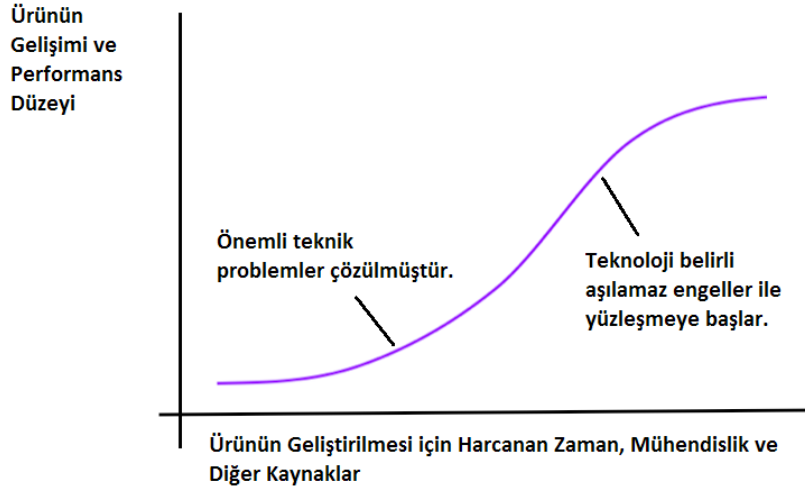
- Teknolojiler de tıpkı ürünler gibi kullanılır, yaygınlaşır ve terk edilirler. Bu durumda teknolojilerin de bir yaşam çevrimi olduğunu kabul etmek gerekir.
- Örneğin, tarihteki ilk otomatik kapı, eski Yunan'da bir tapınak için yapılan, buhar gücü ile çalışan bir kapıdır.
 - Bu kapının mucidi olan rahipler kapıların otomatik açılmasını «mucize» olarak sunarak çıkar etme etmiştir.
 - Ancak günümüzde buhar gücü ile çalışan bir otomatik kapı düşünmek saçmadır.
 - Enerjinin aktarımı ve kullanımı amacını sağlamakta elektrik teknolojileri her türlü mekanik teknolojiye üstünlük sağlamıştır.

Teknoloji Yaşam Çevrimi

- Her teknolojinin yaşam çevrimi farklı olmakla birlikte ortak bir süreç takip edilir.
 - Bu süreci teknolojinin kendisine odaklanan (gelişme, yaygınlaşma, vb) içsel veya kullanıcıların davranışına odaklanan dışsal bir perspektiften ele almak mümkündür.
- Teknoloji yaşam çevrimlerinin hızlı olması ve teknolojilerin hızla terk edilerek farklı teknolojilere geçilmesi o teknolojiyle ilişkili kişilerin davranış biçimlerini önemli ölçüde etkiler.

Teknoloji Yaşam Çevrimi

Teknolojinin Gelişimi Açısından Yaşam Çevrimi

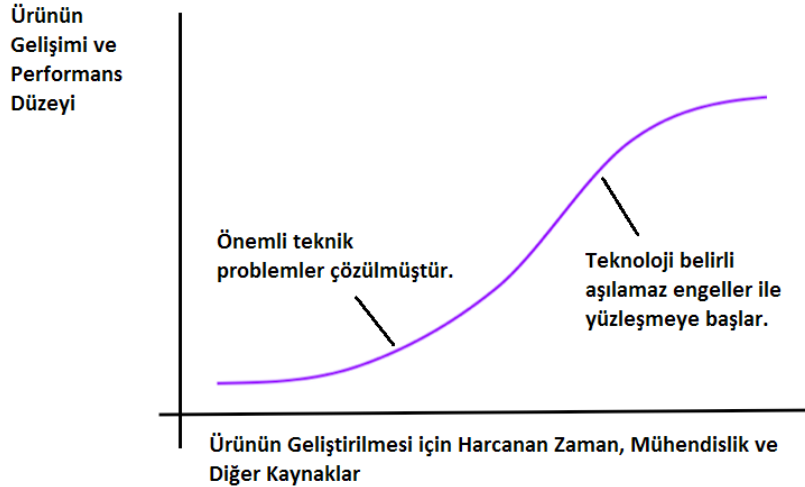


Teknolojinin Yaygınlaşması Açısından Yaşam Çevrimi



Teknoloji Yaşam Çevrimi

Teknolojinin Gelişimi Açısından Yaşam Çevrimi

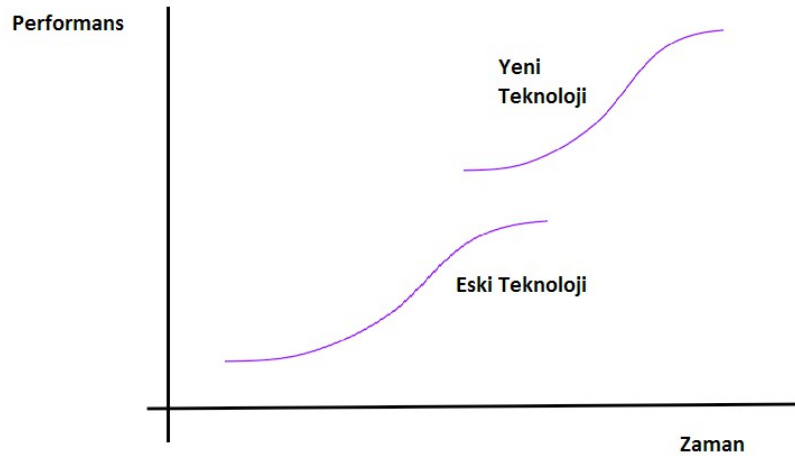


Teknolojinin Yaygınlaşması Açısından Yaşam Çevrimi

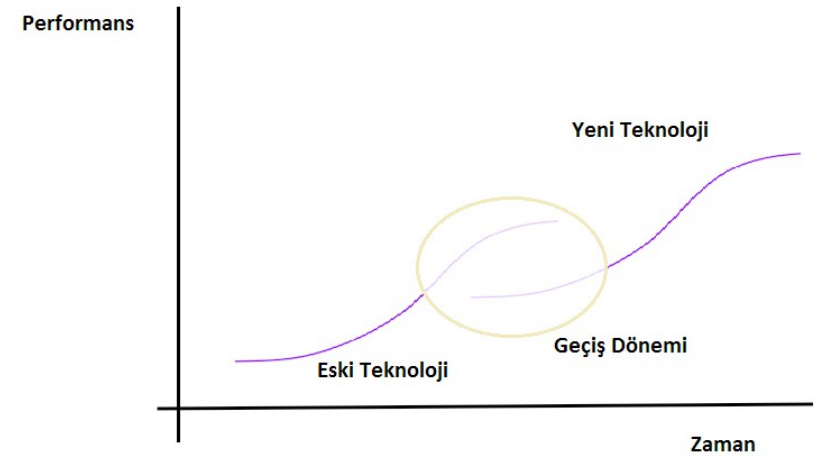


Teknoloji Yaşam Çevrimi

Teknolojinin Hızla Yenilenmesi



Teknolojinin Yavaş Yenilenmesi



Teknoloji Yaşam Çevrimi

- Teknolojiler ürünlerin üretiminde ve kullanımında yer aldığı için teknoloji yaşam çevrimleri ürünlerin yaşam çevrimlerine büyük oranda etki eder. Bu nedenle teknoloji yaşam çevrimleri, ürünlerin kullanıcılarını da etkilemektedir.
 - Öte yandan kullanıcılar genellikle ürünün nihai faydası konusunda hangi teknolojinin kullanıldığının farkında olmazlar.
 - Bu tür detaylara dikkat edenler, teknolojiyi takip eden az sayıda kullanıcı ile başkalarının adına hareket ederek ürün ve teknoloji satın alan, teknik yönü de bulunan kişilerdir.

Teknoloji Yaşam Çevrimi

- Teknoloji ve teknoloji yaşam çevrimleri, özünde kullanıcıların ve üreticilerin teknolojiye verdiği tepkinin açıklanması için geliştirilmiş modellerdir. Yani uygulanmak için geliştirilmiş modeller değildir.
- Öte yandan, bu modelleri iyi anlayan kişilerin, uygun kaynakları da varsa, çevrelerini etkileyerek çevrenin belirli bir teknolojiye olan tepkisini biçimlendirme olanakları mevcuttur.
 - Bu davranış, özellikle küresel ölçekte faaliyet gösteren şirketlerin kabiliyetleri dahilinde olan bir davranıştır ve sık sık gözlenir.
 - Özellikle bilişim teknolojilerindeki büyük şirketlerin müşterilerine «davranış biçimlendirme» amaçlı uygulamaları bulunmaktadır.

Özgür Yazılım

- 1970'lerin sonları ve 1980'lerin başlarında, ticari yazılımların yaygınlaşması ve toplumsal hayatın ayrılmaz bir parçası olması sonucu, yazılımları ve yazılımlar için gereken teknolojileri üreten şirketlerin davranışları «pek de etik olmayan» uygulamaları barındırmaya başlamıştır.
 - Kullanıcıların «kendi gelecekleri üzerine karar verme özgürlüğü»nün tehlikeye girebileceğini öngören bir çok bilim insanı, kendi çevrelerindeki yazılımların geliştirilmesine dair sürecin şirketlerin olumsuz etkilerinden arındırılması için yöntemler geliştirmeye çalışmıştır.
 - Dolayısı ile özgür yazılım hareketi, bilişim teknolojilerinin gelişimi sırasında teknolojinin üretildiği yerin sanayiye geçmesi sonucunda erken dönem teknokritisizm ve teknorealizm düşüncesinin bir eylemi olarak ortaya çıkmıştır.

Özgür Yazılım

Teknorealizm Prensipleri (1998)

- Teknolojiler nötr değildir. Tarafların kendi çıkarları için geliştirilir.
- İnternet devrimsel bir gelişmedir ancak bir ütopyayı hayata geçirmez. Mevcut sosyo-ekonomik sistemin aktörleri İnternet'i kendi amaçları için kullanmaktadır.

Özgür Yazılım

- Tarafların çıkarları arasında uzlaşmanın olabilmesi için ortak geliştirme olanaklarını sağlar. Ancak her zaman için uzlaşamayan tarafların farklı yönlerde gitme (çatallama) özgürlüğü bulunur.
- İnternet, çok sayıda tarafın bir araya gelmesi ve birlikte karar alarak birlikte üretmesi için bir araçtır.

Özgür Yazılım

Teknorealizm Prensipleri (1998)

- Devletler elektronik yaşam biçimleri üzerinde önemli rol oynayacaktır.

Özgür Yazılım

- Devletler özgür yazılımları koruyan temel hukuki prensipleri korumaktadır.
- Ancak aynı zamanda yazılım patentleri, İnternet'in denetimi gibi özgür yazılımlar için «tehlikeli» konular da devletlerin karar alanındadır.

Özgür Yazılım

Teknorealizm Prensipleri (1998)

- Enformasyonun (malumatın) bilgi olmadığı gözardı edilmemelidir. Verilerin ve enformasyonun hızlı dağıtılması, tek başına bilginin ve bilgeliğin yaygınlaşması anlamına gelmez.

Özgür Yazılım

- Enformasyonu hızlı yaymak, örneğin bir uygulama yazılımını kolayca dağıtmak kadar o enformasyondan doğan bilgiyi paylaşmak da önemlidir.
- Özgür yazılım hareketinin faaliyetlerinin önemli bir oranı yazılım geliştirme değil kullanıcı deneyiminin kayda geçirilmesi, paylaşılması ve tartışılması üzerinedir.

Özgür Yazılım

Teknorealizm Prensipleri (1998)

- Okulların veya benzeri kuruluşların teknolojiyi direkt olarak satın alması onları geleceğe hazırlamayacaktır.

Özgür Yazılım

- Özgür yazılımlar ile kuruluşların kendi ihtiyaçları çerçevesinde kendi teknolojilerini üretmeleri ve bu üretimi paylaşımlarının önündeki bazı teknik engeller ortadan kalkmaktadır.

Özgür Yazılım

Teknorealizm Prensipleri (1998)

- Enformasyonun korunması gereklidir. Bu nedenle telif hakları ve benzeri modellerin çağdaş toplumsal ihtiyaçlara göre yeniden düzenlenmesi gerekir.

Özgür Yazılım

- Özgür yazılım lisansları, belirli bir tür enformasyon olan «kod» üzerindeki hak modelini çağdaş ihtiyaçlara göre geliştirmektedir.
- Ayrıca kod dışı varlıklar için de özgür lisans modelleri gelişmektedir.

Özgür Yazılım

Teknorealizm Prensipleri (1998)

- İletişim olanakları ve alt yapıları son tahlilde kamuya aittir bu nedenle bu alt yapılardan öncelikle kamunun fayda elde etmesi sağlanmalıdır.
- Teknolojinin anlaşılması, küresel bir toplumun inşa edilmesi için bir ön koşuldur.

Özgür Yazılım

- İnternet üzerinde özgürlüklerin korunması, özgür yazılımların da hayatta kalması için kritik bir öneme sahiptir.
- Özgür yazılım süreci, kullanıcılarının teknoloji konusunda bilinçlenmesi konusuna ticari yazılım üreticilerinden daha çok önem verir.

Özgür Yazılım

- Özgür yazılım hareketinin ve özgür yazılımların sadece ekonomik işlemler ve sonuçları açısından irdelenmesi, toplumsal faydanın oluşması ve üretim artığının toplum içerisinde dağıtımını konularını öne çıkartmaktadır. Bu nedenle bu bakış açısı ile yapılan çalışmalar çoğu kez özgür yazılımı «sosyalist bir üretim biçimi» olarak sınıflandırabilir.
- Ancak ekonomik faaliyetler sosyolojik çevrenin bir ürünüdür. Aynı şekilde teknoloji de sosyolojik çevre içinde üretilir ve şekillenir.
- Bu nedenle özgür yazılımların içinde yer aldığı üretim ve tüketim biçimi ile özgür yazılımların varlığının etkileri sosyolojik açıdan da değerlendirilmelidir.

Linux

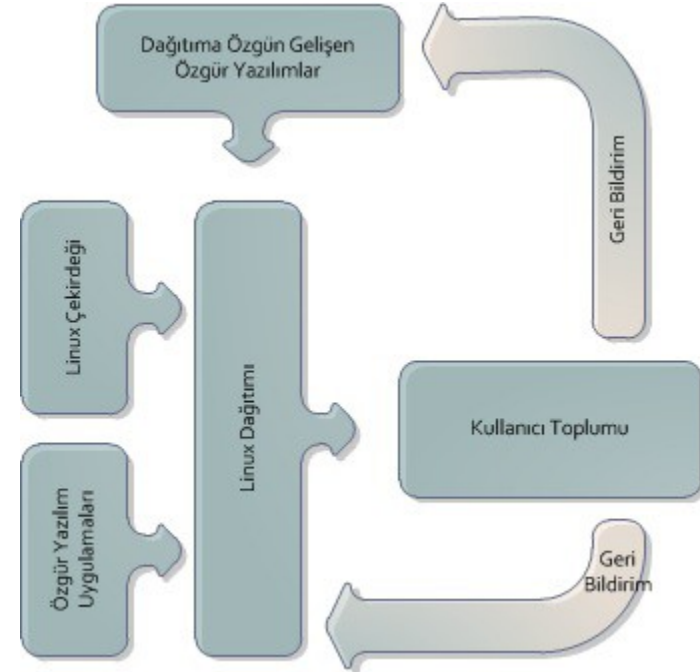
- Linux'un başarısı birden çok faktöre bağlanabilir.
 - Linux, Avrupa kökenli bir yenilik olarak ticari baskının daha az olduğu bir bölgede geliştirilmeye başlanmış daha sonra ise ABD kökenli şirketler tarafından hızla ticari ihtiyaçlara da uyan bir hale getirilmiştir.
 - Linux, özgür yazılım toplumunun çalışma esaslarının ortaya çıktığı ve belirli düzeye geldiği bir dönemde geliştirilmeye başlamıştır.
 - Linux bu döneme kadar varlığı bilinen ancak doldurulamayan bir boşluğu doldurmak için geliştirilmiştir.
 - Linux'un geliştirilmesinde görev alan kişiler «yaptıkları işe dayalı» bir eleme sürecinden geçmektedir.
 - Linux'un ne yöne doğru gelişeceği konusu bir çok kişinin farklı ortamlarda elde ettiği deneyimlerden gelen görüşlerin sentezi ile karar verilen bir konudur.

Linux

- Linux'un başarısının sayılan faktörlerinin hepsi de bir ticari işletme için de geçerli olabilecek faktörlerdir. Ancak Linux'un uzun süre devam eden bir başarıya sahip olmasının nedeni Linux toplumunun ortak bir amaç için bir araya gelen alt topluluklardan oluşmasıdır.
 - Linux çekirdeği projesinin amacı, özgür yazılım topluluklarının özgür yazılımları istedikleri donanım üzerinde herhangi bir sınırlama olmadan kullanabilecekleri bir işletim sistemine sahip olmasıdır.
 - Linux dağıtım projelerinin amaçları ise, kendileri için uygun gördükleri bir kullanıcı kitlesine, o kitle için öncelikli gördükleri bir dizi niteliğe sahip bir özgür yazılım seçkisini, genişletilebilir bir çerçeve içinde sunmaktır.

Linux

- Günümüzde belirli bir yaygınlığa sahip tüm Linux dağıtımları ana görevleri diyebileceğimiz, özgür yazılımlar ile kullanıcılar arasında «dağıtım kanalı» olma işini başarı ile yerine getirmektedir.



Linux

- Bununla birlikte, dağıtımın kullanıcılarının (hedef kitlesinin) hangi tür özgür yazılımları ne şekilde kullanmak istediğine bağlı olarak önemli değişiklik gösterebilecek olan iki faktör söz konusudur:
 - Özgür yazılımların dağıtımın içerisinde bir araya getirilme biçimi
 - Kullanıcıların bir veya birden çok sistemde kurulu olan yazılımların kurulum, kaldırma ve güncelleme işlemlerini sistematik olarak yönetme konusunda tercih ettikleri yöntemler
- Bu iki konu da «yazılım yapılandırma yönetimi» (İng. Software configuration management) adı verilen önemli bir teknik problemin gündelik yaşama yansımalarıdır.
 - Bulut bilişim uygulamalarının belirli düzeyde yaygınlaşması, bu problemi daha farklı bir düzeye taşımaktadır.
 - Türkiye’de bulut bilişimin emekleme döneminde olması bir sorun olmakla birlikte, bu problemin ötelenmesine de neden olmaktadır.

Linux

- Her Linux dağıtımı, kendi hedef kitlesinin yazılım yapılandırma yönetimi ihtiyaçlarını etkili biçimde çözmek ve bunu kullanıcıların kolay ve verimli çalışmasını sağlayacak biçimde çözmek zorundadır.
- Bu nedenle dağıtımların diğerlerinden en büyük farkı bu sorunu çözen «paket yönetim sistemi» tercihlerinde farklıdır.
 - Dünya çapında ilk ortaya çıkan ve hala en yaygın kullanılan iki paket yönetim sistemi olan APT (Debian) ve RPM (RedHat) sistemleridir.
 - Ancak bununla birlikte, zaman zaman farklı amaçlara hitap edebilecek farklı paket yönetim sistemleri de üretilmektedir.
 - Ayrıca paket yönetim sistemlerinin gelişiminde kullanılabilecek yan teknolojiler (ör. Torrent protokolü) üretilmektedir.

Linux

- Linux dağıtımlarının ayrıştığı ikinci bir nokta, yine hedef kitlelerinin ihtiyaçlarına uymak adına dağıtıma dahil ettikleri özgür yazılımların tercih edilen sürümleri konusundaki tercihleridir.
 - Dağıtımlar bu tercihlerini kendi paket depolarına yerleştirdikleri dağıtım sürümleri yolu ile gösterirler.
 - Hiç bir testten geçmeyen bir paketin dağıtımın deposunda yer bulması, kullanıcılar açısından geriye dönüşü zor sorunlara yol açabilir ve dağıtımın prestij kaybına yol açar.
 - Bu nedenle, her bir dağıtımın paket deposuna girecek olan bir uygulama veya güncellemenin geçirmesi gereken işlemler konusunda bir süreci bulunur. Bazı dağıtımlar için bu süreçler kapsamlıdır bazıları için ise oldukça basittir.

Linux

- Bir dağıtımın deposuna eklediği tüm yazılımların dağıtım ve biri biri ile olan etkileşimlerinde «beklenildiği gibi davranma» niteliğine «kararlılık» (İng. Stability) adı verilir.
 - Kararlı bir dağıtım ve o dağıtımın deposunda bulunan yazılımların herhangi bir kombinasyonunun kullanıcıyı şaşırtmayacak bir şekilde çalışacağı öngörülmektedir.
- Linux dağıtımları, kararlılık konusunda değişen derecede hassas hedef kitleler için hazırlanır.
 - Ev kullanıcıları
 - Küçük işletmeler
 - Büyük işletmeler
 - Kamu kurumları
 - Görev kritik sistemler (Ör. Ulaştırma, sağlık, savunma sanayi)

Linux

- Linux dağıtımlarının bazıları kullanıcı arayüzü konusunda zengin bir deneyim sunmayı hedefler. Bazı dağıtımlar için ise bu önemli bir konu değildir.
- Kullanıcı arayüzü deneyiminin oluşturulmasında önemli bileşenler masaüstü yöneticisi, temalar ve arayüz içerisinde çalışan uygulamaların bütünleştirme (İng. integration) düzeyidir.
 - Masaüstü yöneticileri çok hızlı biçimde geliştirildiği için son sürümleri kararlılık açısından önemli riskler içerir ve bir çok dağıtım tarafından bilinçli olarak kullanılmaz.
 - Tema geliştirmek zor olmayan bir iş olmakla birlikte bir çok farklı yazılımın o temadaki kullanım sırasında nasıl görüneceğinin test edilmesi zaman almaktadır.
 - Bütünleştirme noktalarının tespit edilmesi ve bütünleştirmenin yapılması önemli bir emek gerektirir. Ayrıca tek tek her bir bağımsız yazılımın güncellemelerinin de bütünleştirmeye uygun biçimde yapılması gerekeceği için uygulamaların paketlerinin hazırlanması da önemli bir emek gerektirmektedir.

Linux

- Linux dağıtımları hedef kitlesi belirlerken, belirli bir işi yapan, belirli bir kurumda çalışan, aynı dili konuşan yada aynı bölgede yaşayan kişiler gibi kriterlere göre de odaklanabilir.
 - Bu durumda, odaklanılan kitlenin ana ihtiyaçlarının eksiksiz biçimde karşılanması esastır.
- Türkiye'ye özel bir dağıtımın hedef kitlesinin tartışılmaz ihtiyaçlarından birisi erişmekte ve anlamakta güçlük çektiği İngilizce kaynakların yerine geçecek Türkçe içeriğin sağlanmasıdır.
 - Öte yandan içeriğin salt Türkçe olması yeterli değildir.
 - Kullanıcı ihtiyaçlarına uygun, kolay anlaşılır ve uygulanır, nitelikli ve doğrulanabilir bir içeriğin oluşturulması gereklidir.

Linux

- Bir Linux dağıtımının hedefi netleştikten sonra, kullanıcı toplumunun dağıtıma olan katkısını artıracak süreçleri kurması gereklidir.
 - Deneyimi aktaran belgelerin oluşturulması ve kalite güvencesinin sağlanması,
 - Çok dil desteği için çeviri ve çeviri kalite güvencesi,
 - Paket oluşturulması,
 - Çeşitli düzeyde test ve kalite güvence ekiplerinin kurulması ve yönetilmesi,
- Bunlara ek olarak kullanıcı toplumu ve birlikte çalışılan üçüncü organizasyonlar (örneğin bazı yaygın özgür yazılımların projeleri) ile «geliştirme» konusunda işbirliklerinin de sağlanması gereklidir.
 - Kitaplıkların ortak geliştirilmesi veya ortak kullanımı,
 - Sistem bütünleştirilmesine dönük ortak testler,
 - Ortak iş ve sürüm takvimi,

Linux

- Bir Linux dağıtımının, hedef kitlesinin kullanım biçimine dair alışkanlıklarını iyi incelemesi ve bu kitleye yararlı olacak ek uygulama yazılımları geliştirerek kullanıcıya sunduğu değeri artırması beklenir.
 - Örneğin, İnternet bağlantısı zayıf olan ve güncellemeleri İnternet'den indirmekte güçlük çeken kişilerin ağırlıklı olarak kullanacağı bir dağıtımın, CD veya USB bellek türü ortamları kullanarak güncelleme yapma kabiliyeti kazanması gerekir.
- Ayrıca her bir Linux dağıtımının, diğerlerine ait yenilikleri incelemesi ve özgür yazılım prensipleri çerçevesinde kendisine entegre olması da beklenir.

Linux

- Linux'un geliştirme ve dağıtım modeli bir çok uygulama alanı için önemli üstünlükler sağlamakla birlikte, bazı alanlarda ancak kısmen etkili olabilmektedir.
- Bu alanlardan birisi bilgisayar oyunu alanıdır.
 - Günümüzde bilgisayar oyunları bir uygulama yazılımından çok bir sinema filmi gibi üretilmektedir. Üretici firmalar kendilerini «prodüktör» olarak adlandırmaktadır.
 - Bilgisayar oyunlarının albenisi olan grafik ve müzik içeriği tek kullanımlık üretildiği için «yeniden kullanım» olanakları düşüktür.
 - Ayrıca kullanıcıların bir oyunu hızla «tüketip» bir sonrakine geçmesi, oyunların sonraki sürümlerinin aslında sıfırdan yazılmış bambaşka oyunlar olması da kullanıcı geribildiriminin getirebileceği avantajları arka plana itmektedir.
 - Bu nedenle Linux ve özgür yazılım süreçleri, oyun sektöründe nihai ürünlerden çok oyun motorları gibi arka planda çalışan ortak sistem yazılımları için işe yaramaktadır.

2. Bölüm Linux'un Detayları

- Bu bölümde, Linux'un bir işletim sistemi olarak yapısı incelenecektir.
- Bu kapsamda, sistem açılış süreci, aygıtlar ve dosya sistemi konuları işlenecektir.

Sistem Açılış Süreci

- Linux sistemlerin açılış süreci genellikle Intel x86 ve takip eden mimarilere sahip sistemler üzerinde anlatılmaktadır.
 - Burada da bu yaklaşım takip edilecektir.
 - Bununla birlikte, açılış sürecinin hangi adımlarının donanım mimarisi ile bağlantılı olduğunun göz ardı edilmemesi gereklidir.
- Donanım açısından bakarsanız, Intel mimarili sistemlerde açılış süreci «BIOS'dan işletim sistemi çekirdeğine akan» bir «denetim devri» sürecidir. Burada denetimden kast edilen şey, işlemciye yüklenecek olan kodları kimin belirlediğine dair denetimdir.

Sistem Açılış Süreci

- Linux yükleme mekanizması yandaki şekilde sıralanan adımlardan oluşmaktadır.



Sistem Açılış Süreci

BIOS

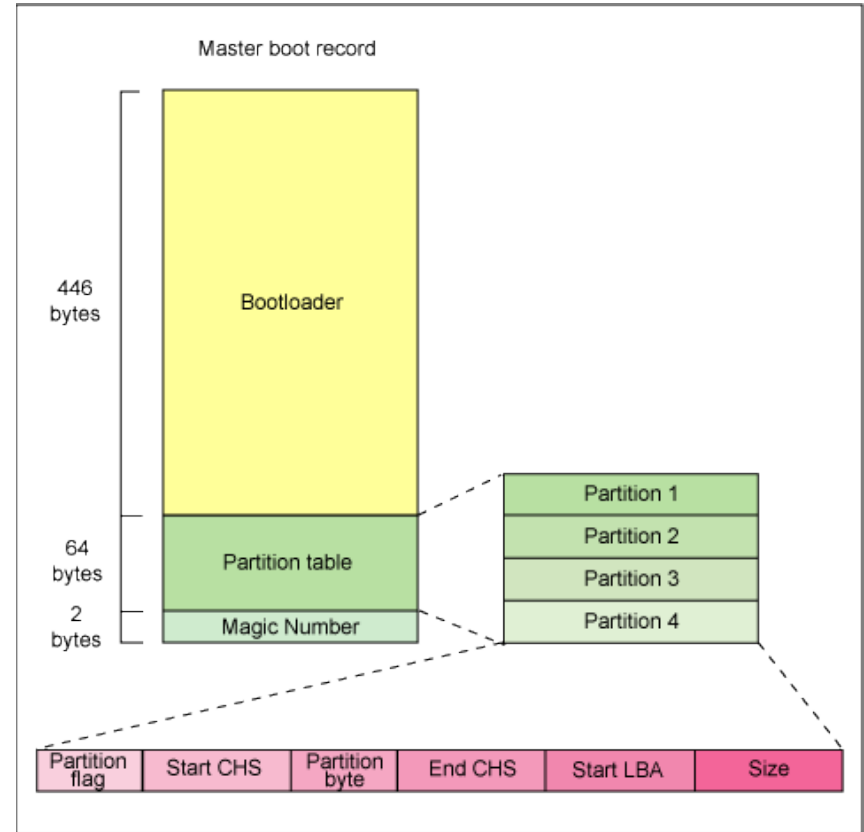
- 1970'lerde tasarlanmıştır.
- Sistem aygıtlarında temel kontroller yaptıktan sonra kendi içerisindeki bellekte kaydedilen sıra ile «açılış aygıtlarını» bir ana yükleme kaydı (İng. Master Boot Record, MBR) için kontrol eder.
- İlk bulduğu MBR'yi belleğe yükler ve çalıştırır.

UEFI

- 1990'larda tasarlanmaya başlanmış, güncel halini 2008 yılında almıştır.
- Intel tarafından geliştirilmeye başlanmıştır ancak günümüzde Intel dışı mimarilerde de kullanılabilir.
- Açılış süreci sonrasında çalışma zamanında da bazı hizmetler sunabilmektedir.
- Açılış işlemleri daha detaylı olmakla birlikte BIOS süresine benzemektedir.

Sistem Açılış Süreci

- Bir depolama aygıtının başında bulunan MBR, 512 bayt uzunluğunda küçük bir alandır.
 - Günümüzdeki gelişmiş yükleyicilerin ve ayar dosyalarının bu alana sığması mümkün değildir.
 - Bu nedenle MBR, günümüzde «Adım 1» (İng. Stage 1) yükleyiciyi barındırır.
 - Adım 1 yükleyici, Adım 2 yükleyicinin bir disk üzerindeki konumunu bilir ve bu konumdan Adım 2 yükleyiciyi çalıştırır.



Sistem Açılış Süreci

- Adım 2 yükleyici, işletim sistemi çekirdeğini yüklemekle görevlidir. Bu nedenle Adım 2 yükleyicinin kabiliyetlerinin daha gelişmiş olması gerekir.
- Günümüzde yaygın kullanılan Adım 2 yükleyici GRUB'dır.
 - GRUB'ın en önemli kabiliyeti Linux dosya sistemlerini tanımasıdır.
 - Adım 2'den hemen önce (Adım 1,5 denebilir) çekirdeğin yükleneceği disk bölümüne ait dosya sistemi bağlanır. Böylece GRUB çekirdek ve bir dizi farklı dosyayı okuyabilir.
 - Adım 2'de ise yüklenebilecek olan alternatif çekirdeklerin bir listesi ekrana getirilerek kullanıcı etkileşimi yolu ile listeden bir seçim yapılması sağlanır.
- GRUB'ın alternatif çekirdek listesi `/etc/grub.conf` dosyasında yer alan ayarlarda belirlenmektedir.
 - GRUB, Linux çekirdeğini yüklerken geçici olarak bellekte yer alan bir RAM sürücüsü (İng. RAM Drive) oluşturur.
 - GRUB, Linux dışı işletim sistemlerini de yükleyebilir.

Sistem Açılış Süreci

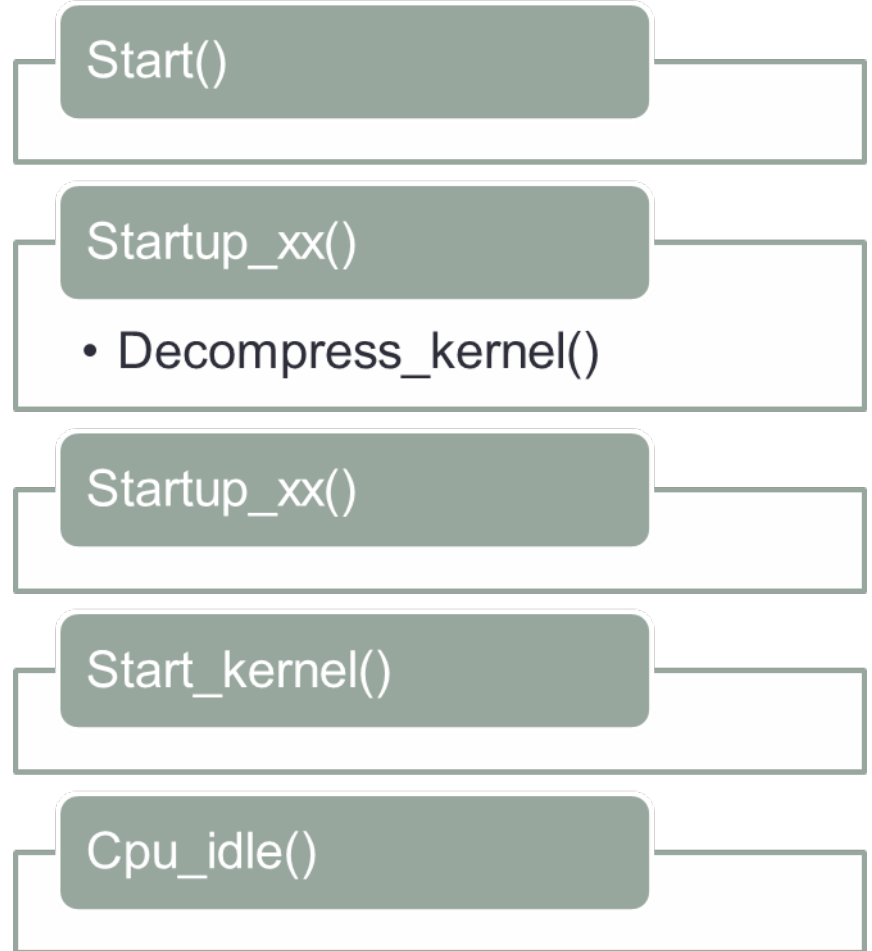
- GRUB'ın dosya sistemlerini bağlayarak istenen bir yerden dosya okuyabilmesi, GRUB'ın değişik kabiliyetler kazanmasına neden olmuştur.
 - Örneğin GRUB ekranı renkli bir arka plan resmi ile güzel görünür hale getirilebilir.
 - Bir diğer örnek olarak yüksek güvenli sistemlerde kullanılan eklentilere sahip bir GRUB sürümü (Trusted GRUB), yükmeden önce işletim sistemi çekirdeğinin değiştirilip değiştirilmediğini kontrol edebilmektedir.
 - Öte yandan, GRUB bir işletim sistemi olmadığı için kullanıcıya sunabileceği etkileşim sadece klavye ile sınırlıdır. Fare desteği yoktur. Buna ek olarak ses dosyası oynatma, vb çoklu ortam becerileri de yoktur. Açılış sırasında ses duyulması için işletim sistemi çekirdeğinin yüklenmesi beklenmelidir.

Sistem Açılış Süreci

- GRUB, kendisinden önce yaygın kullanılan LILO yükleyicisinin yerine geçtikten sonra uzun süre boyunca küçük özellikler eklenerek gelişmiştir.
 - Muhafazakar bir geliştirici ekibine sahiptir.
 - Bir çok Linux dağıtımı GRUB'ın 0,97 sürümü ile çalışmaktadır – ki 0,97 dahi son derece olgun bir sürümdür.
 - GRUB'ın 1.x sürümleri ve Haziran 2012'de çıkan 2.0 sürümü, 2002 yılından bu yana Japon Bilim ve Teknoloji Bakanlığı'nın Bilgi Teknolojileri Yaygınlaştırma Ajansı önemli katkıları ile geliştirilmektedir.
 - Ubuntu 2009, Fedora 2011, OpenSuse 2012 yılında GRUB 1.x serisini varsayılan yükleyici olarak kullanmaya başlamıştır.
- GRUB'ın en önemli özelliklerinden birisi disk MBR'sine yazıldığı sırada bir işletim sistemi bulunmasına gerek olmamasıdır. Böylece işletim sistemi kurulmamış ancak GRUB kurulu bir ön sistem hazırlanabilir.

Sistem Açılış Süreci

- GRUB tarafından belleğe yüklenen Linux çekirdeği esasında sıkıştırılmış bir çekirdek imajıdır. Tipik olarak 512 KB büyüklüğünde olan bu imajın açılması gerekir.
 - Sıkıştırılmış imaj temel bazı donanım kontrolleri yaptıktan sonra esas çekirdeği yüklemeye başlar.
 - Yüklenen çekirdek, çekirdeğin açılış sürecini başlatır.
- Çekirdek kendi açılışını bitirdikten sonra «kullanıcı uzayındaki» ilk uygulama olan init betiklerini çalıştırır.



Sistem Açılış Süreci

- Linux işletim sisteminin açılışında en çok sayıda işlemin yapıldığı adım Init() sürecidir.
- Bu sürecin işlerilmesi için birden fazla alternatif bulunmaktadır. Dağıtımlar, son yıllarda bu alternatifler arasında geçiş yapmaya başlamıştır.
 - SysV Init süreci, geleneksel UNIX mimarisindeki bir süreçtir ve en yaygın kullanılan süreç olarak görülmektedir.
 - Upstart init süreci, 2006'dan bu yana Ubuntu için geliştirilen bir süreçtir.
 - Systemd init süreci, bağımsız bir yapıda geliştirilmiştir ve 2010 yılından sonra bir çok dağıtım tarafından kullanılmaya başlanmıştır.
 - Müdür init süreci Pardus 2000 serisi için yazılmıştır. Yapı olarak Upstart'a benzemektedir. Ancak geliştirilmesi düzenli olamamıştır.
- SysV init süreci, diğerleri için referans noktası teşkil etmekte olduğundan dolayı öncelikle SysV Init sürecinin anlatılması daha sonra ise Upstart ve Systemd süreçlerinin ek özelliklerinin tartışılması tercih edilmektedir.

Sistem Açılış Süreci

- Init() süreci, işletim sistemi çekirdeği tarafından çalıştırılan ilk süreç olduğu için süreç tanımlayıcısı (pid) 1 olacaktır.
- SysV init süreci açılış sonrasında Linux'un hangi «çalışma seviyesinde» çalışacağına karar verilmesi ve seçilen çalışma düzeyi için gereken bileşenlerin yüklenmesi olarak özetlenebilir.
 - Süreç içinde bu karar /etc/inittab dosyasına bakarak verilir.
 - Toplam 7 çalışma düzeyi (seviyesi, vs) bulunur.
 - Her bir çalışma düzeyi içinde yüklenmesi gereken bileşenler içinde çalıştırılması gereken uygulamalar da bulunacağı düşünülmelidir.
 - Bu programların sistem yönetici tarafından kolayca düzenlenmesi amacı ile değişik bir düzen oluşturulmuştur.

Sistem Açılış Süreci

- Çalışma düzeyleri
 - 0 – Sistemi kapat
 - 1 – Tek kullanıcı
 - 2 – Çok kullanıcı, NFS yok
 - 3 – Eksiksiz çok kullanıcı
 - 4 – Boş bırakılmıştır
 - 5 – X11
 - 6 – Yeniden başlat
- Çalışma düzeyi betiklerinin yerleri
 - /etc/rc.d/rc0.d/
 - /etc/rc.d/rc1.d/
 - /etc/rc.d/rc2.d/
 - /etc/rc.d/rc3.d/
 - /etc/rc.d/rc4.d/
 - /etc/rc.d/rc5.d/
 - /etc/rc.d/rc6.d/

Sistem Açılış Süreci

- SysV init() süreci, rc*.d dizinlerindeki betiklerin doğru sıra ile çalıştırılmasına dayanmaktadır.
 - Sistem yöneticiler her bir betiğin çalıştırdığı yazılımın diğer yazılımlar ile olan bağımlılıklarını takip ederek önce hangi yazılımın çalıştırılacağına dair planlama yaparak betik numaralarını oluşturur.
 - Öte yandan yaygın yazılımların başlatma ve kapatma betiklerini zaten o yazılımları hazırlayanlar yazar. Bu betikler bilinen tipik bağımlılıkları göz önüne alarak hazırlanmıştır.
- Sistem yöneticisinin betiklerin içeriği veya sırası ile oynaması yada sırada araya fazladan bir betik eklemesi mümkündür. Böylece sistem üzerinde ince ayar yapılabilir.
- SysV init() süreci, açılış sırasında en çok zaman alan işlem adımıdır.

Sistem Açılış Süreci

Systemd

- SysV init() sürecinin geliştirilmesini hedef almıştır.
 - Daha çok paralel işlem.
 - Servisler arası bağımlılıkları açıkça ifade etmek için bir mekanizma.
 - Kabul kitaplıkları kullanımını azaltmak.
 - Servislerin soket veya veriyolu (İng. Bus) temelli çalıştırılması.
 - Süreç ID'leri yerine süreç gruplarının izlenmesi yolu ile gruplama.
- Sadece Linux'a dönük tasarım.
- SysV init() sürecine uyumludur.
- Udev alt yapısını içermektedir.

Upstart

- Sistem açılış sürecinin çok daha dinamik hale gelmesini hedef almıştır.
- Ancak SysV betiklerini çalıştırabilir.
- Olay tabanlı modeli takip ettiği için üstünlükleri vardır.
 - Eşzamanlı ve «bloklayan» işletim zorunluluğu yoktur.
 - Görevlerin çalıştırılıp çalıştırılmayacağına «o anda karar verme» olanağı vardır.
 - Sistem açılışı sonrasındaki olaylar (örneğin USB olayları) kolay yönetilebilir.
- Orta-uzun dönemde cron, anacron ve atd görevlerini de yönetmesi öngörülmektedir.

Sistem Açılış Süreci

Systemd

- Gnome tarafından önerilen init() süreci.
- Fedora, Mandriva ve OpenSuse'nin varsayılan init() süreci.
- RHEL 7 için değerlendirme sürmektedir.
- Arch, Debian ve Gentoo'da istenirse kurulabilen alternatif init() süreci.

Upstart

- Ubuntu 6.10'dan bu yana upstart kullanmaktadır.
- RHEL 6 sürümü ve dolayısı ile CentOS, Oracle Linux, vb dağıtımlar da upstart kullanmaktadır.
- Fedora 9-14 sürümlerinde kullanılmıştır. 15 sürümünde systemd kullanılmıştır.
- OpenSuse 11 sürümünde kullanılmıştır. 12 sürümünde systemd kullanılmıştır.
- Maemo (Nokia), ChromeOS, vb daha az yaygın dağıtımlarda da upstart kullanılmaktadır.

Sistem Açılış Süreci

- Görüldüğü üzere upstart ve systemd iki önemli alternatif olarak karşımıza çıkmakla birlikte, SysV init() sürecine göre tasarlanan açılış betikleri hala kullanılabilir.
- Pardus 2000 serisi ile tasarlanan ancak sürdürülemeyen Müdür init() yönetim süreci, tasarım itibarı ile Upstart'a benzemektedir ancak Upstart'ın bazı özelliklerinden yoksundur. Geliştirilmesinin sürdürülmesi durumunda Upstart'ın mevcut yapısına benzer bir yapıya ulaşacağı öngörülmektedir.

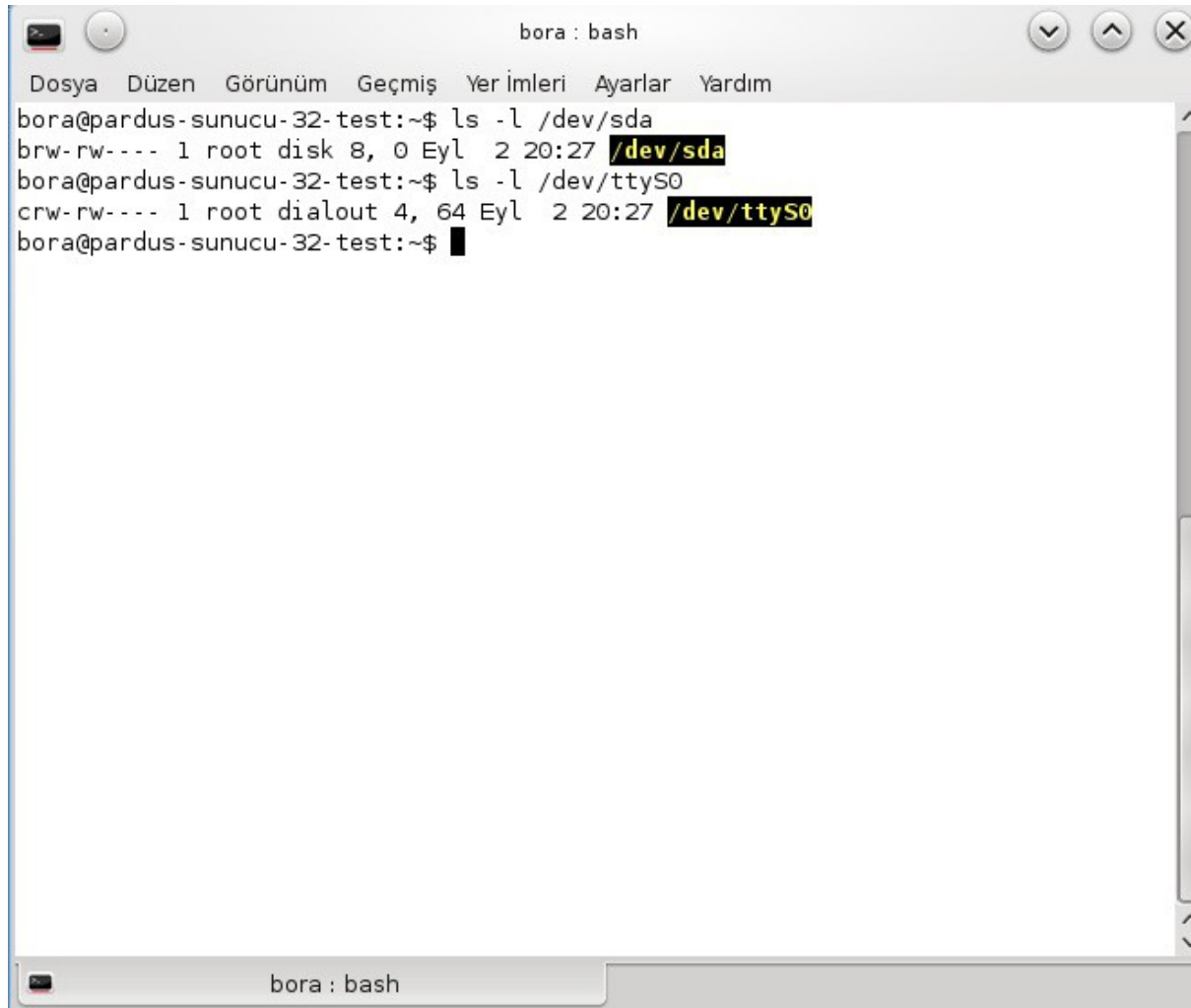
Aygıtlar

- Linux aygıtlara erişimde UNIX modelini takip eder. Bu modelde her bir aygıt için bir adet «temsili dosya» bulunmaktadır ancak bu dosyaya erişim hakları ve erişim biçimi konusunda çok sayıda kısıtlama bulunmaktadır.
 - Linux aygıt sürücüleri, aygıtta ait orijinal «temsili dosyaya» olan erişimi düzenlemek için yazılır ve kullanıcıya aygıtı farklı şekillerde kullanmak için farklı «temsiller» sağlarlar.
 - Bu nedenle Linux'da aygıt sürücüsü yazmak veya yüklemek aygıt üreticisi için daha zor olmakla birlikte, bir aygıt sürücüsüne ince ayar yapmak daha kolay olduğundan dolayı aygıt sürücülerini hazırlayan üreticiler uzun dönemde daha avantajlı hale gelir.
- Kullanıcı açısından bakıldığında ise, aygıt sürücüsü çekirdeğe bir modül olarak yüklenmiş ise o aygıtta erişmek mümkündür.
 - Linux'da kullanıcının (yada sistem yöneticisinin) aygıt sürücüleri ve bu sürücüler arasındaki etkileşimi düzenlemesi için çok sayıda komut satırı aracı bulunmaktadır.

Aygıtlar

- Aygıtlar söz konusu olduğunda sıklıkla karakter aygıtları ve blok aygıtlardan bahsedilir.
 - Karakter aygıtları, verilerin ardışık olarak okunduğu aygıtlardır. Dolayısı ile okuma sırsında rastgele konumlara geçmek mümkün değildir. Bir çok karakter aygıtı (örneğin klavye) aynı zamanda bir seferde tek bir karakterin okunduğu aygıtlardır ancak bu bir zorunluluk olmaz.
 - Blok aygıtları, verilerin ardışık olarak okunmasının zorunlu olmadığı aygıtlardır. Veri bu aygıtın temsil ettiği alanın herhangi bir yerinden bir blok olarak okunur. Dolayısı ile rastgele atlamalar mümkündür. Diskler tipik birer blok aygıttır.
- Bu iki türe ek olarak «ağ aygıtı» (İng. Network device) olarak adlandırılan üçüncü bir aygıt türü daha vardır. Bu aygıtlara ağ arayüzleri ile erişilir.

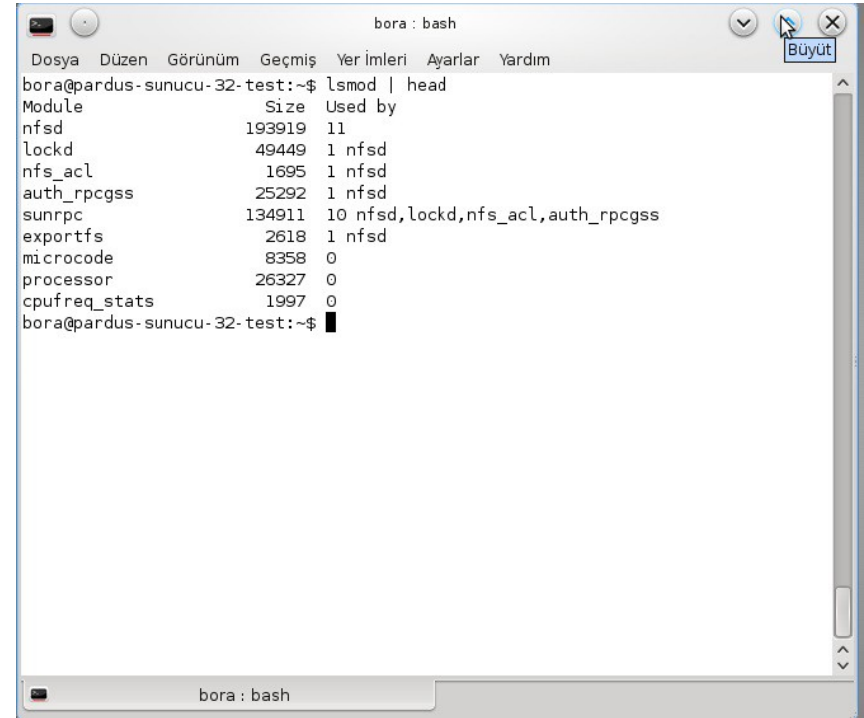
Aygıtlar



```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ ls -l /dev/sda
brw-rw---- 1 root disk 8, 0 Eyl  2 20:27 /dev/sda
bora@pardus-sunucu-32-test:~$ ls -l /dev/ttyS0
crw-rw---- 1 root dialout 4, 64 Eyl  2 20:27 /dev/ttyS0
bora@pardus-sunucu-32-test:~$
```

Aygıtlar

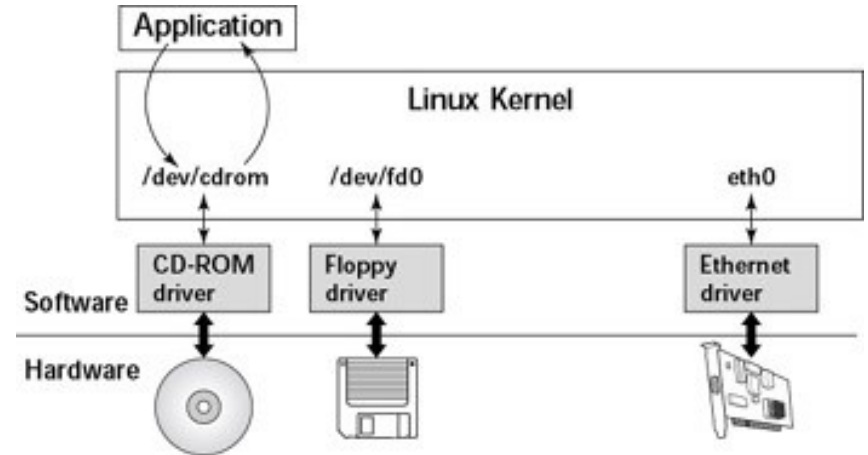
- Linux yüklü bir bilgisayar kapalı iken o bilgisayarda bulunan aygıtlar tamamen değiştirilse dahi, Linux o aygıtlar için uygun aygıt sürücülerini arayıp yükleyerek sistemi başarı ile açabilir.
- Bu derece dinamik bir sistemin çalışabilmesi için aygıt sürücüler yüklenirken kendilerini çekirdeğe kaydettirir.



```
bora@pardus-sunucu-32-test:~$ lsmod | head
Module                Size Used by
nfsd                   193919 11
lockd                   49449 1 nfsd
nfs_acl                 1695 1 nfsd
auth_rpcgss            25292 1 nfsd
sunrpc                 134911 10 nfsd,lockd,nfs_acl,auth_rpcgss
exportfs               2618 1 nfsd
microcode              8358 0
processor              26327 0
cpufreq_stats         1997 0
bora@pardus-sunucu-32-test:~$
```

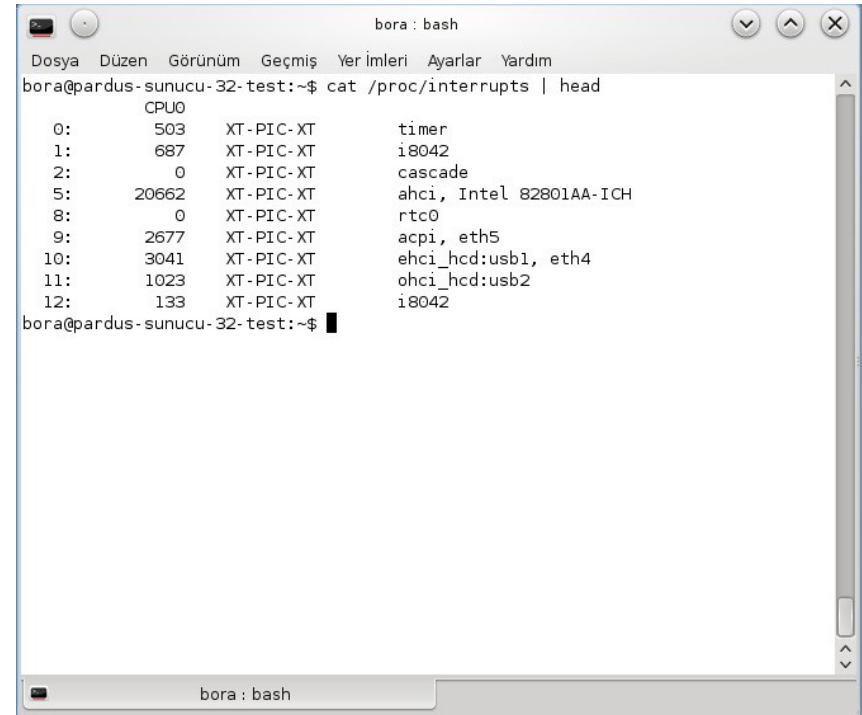
Aygıtlar

- Linux'da aygıt sürücülerinin temel işlevi, aygıtların yapısını uygulamalardan soyutlamak ve aygıtı dış dünyaya bir dosya gibi göstermektir.
- Böylece dosyaya yazma ve dosyadan okuma ile ilgili bir çok temel programlama alt yapısı kullanılarak aygıta erişen uygulama yazılımları yazılabilir.



Aygıtlar

- Aygıt sürücüleri temsil ettikleri aygıtların durumlarını takip etmek durumundadır. Bu takip için iki alternatif yöntem bulunur.
 - Bloklama : Bu yaklaşımda sürücü tüm çekirdek işlemlerini bloklar ve aygıtı sürekli olarak tarayarak bekler. Çok özel durumu olan bazı cihazlar (Ör. Kripto alt sistemleri) hariç hiç bir cihazın bu şekilde kullanılması tercih edilmez.
 - Kesme : Bu yaklaşımda sürücü çekirdek işlemlerini bloklamaz ancak aygıtın işlem gerektirip gerektirmediği çekirdek tarafından izlenir. Bu tür bir ihtiyaç ortaya çıktığında «kesme» (İng. Interrupt) gerçekleşmiş olur.



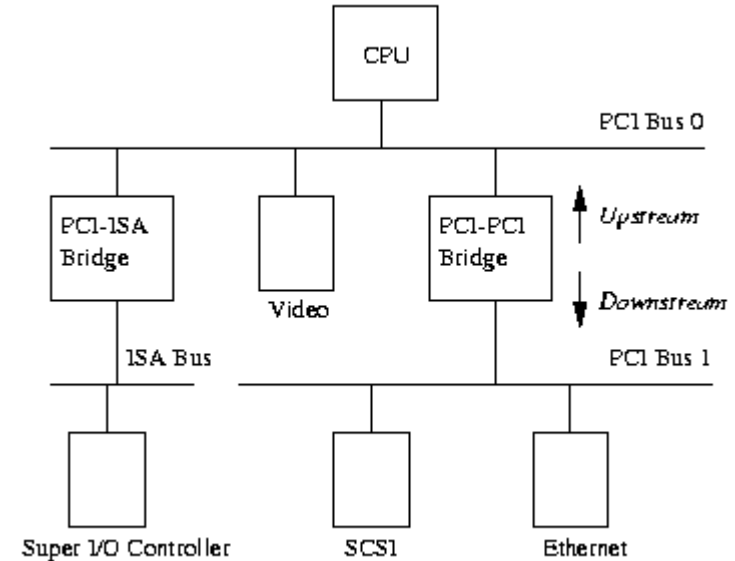
```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer imleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ cat /proc/interrupts | head
CPU0
0: 503 XT-PIC-XT timer
1: 687 XT-PIC-XT i8042
2: 0 XT-PIC-XT cascade
5: 20662 XT-PIC-XT ahci, Intel 82801AA-ICH
8: 0 XT-PIC-XT rtc0
9: 2677 XT-PIC-XT acpi, eth5
10: 3041 XT-PIC-XT ehci_hcd:usb1, eth4
11: 1023 XT-PIC-XT ohci_hcd:usb2
12: 133 XT-PIC-XT i8042
bora@pardus-sunucu-32-test:~$
```

Aygıtlar

- Kesme güdümlü bir sistem, aygıtlardan alınan verinin hızı düşükken çok kolay biçimde çalıştırılabilir. Ancak günümüzde bazı aygıtların (özellikle diskler ve ağ kartları) transfer edebilecekleri veri çok büyük miktarlardadır ve işlemcinin veri işleme kapasitesi ile yarışmaktadır.
 - Doğrudan bellek erişimi (İng. Direct Memory Access – DMA) bu amaçla geliştirilmiş bir tekniktir.
 - DMA ile bir aygıt veriyi doğrudan belleğe yazabilir.
 - DMA'nın anakart üzerinde ayrı bir denetleyicisi bulunur ve DMA işlemleri sınırlı sayıda kanal üzerinden gerçekleştirilir.
- Aygıt sürücülerinin DMA kanallarını kullanması durumunda bu durumun yönetilmesi gerekir. Linux çekirdeği DMA erişimini de yönetir.
 - Bazı aygıt sürücülerini sürekli olarak aynı DMA kanalını kullanırken bazıları farklı kanalları kullanabilir.

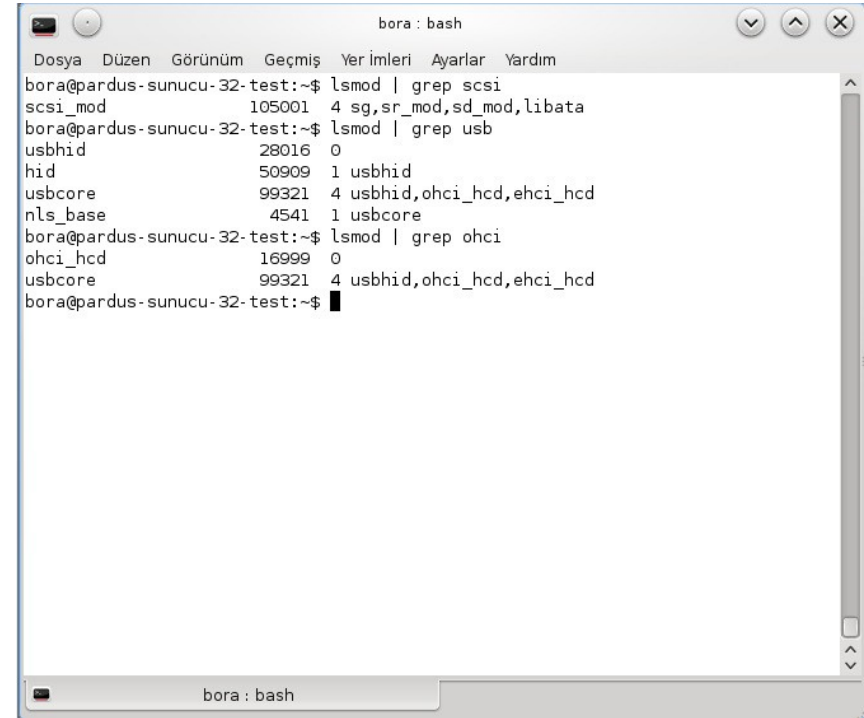
Aygıtlar

- PCI aygıtların davranışları klasik aygıtlardan daha gelişmiştir.
 - Her bir PCI aygıtının sistem bellek adreslerinden bağımsız bir adres uzayı bulunur.
 - PCI aygıtları kendi bellek uzaylarına yazarken sistemin normal belleği ile çalışma olmadığı için çekirdeğin süreçleri durdurması gerekmez.
 - PCI aygıtının belleğindeki verilerin normal belleğe aktarılması gerekince DMA kullanılır.



Aygıtlar

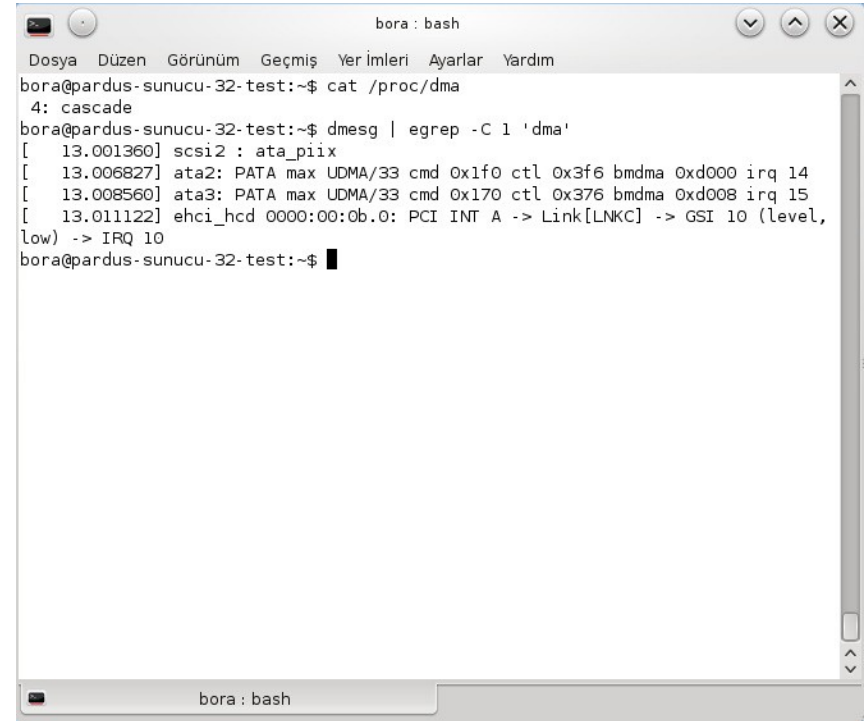
- SCSI ve USB aygıtların yapısı da PCI aygıtlar gibi iki aşamalıdır.
 - «SCSI Host Controller» ve «USB Host Controller» adını verdiğimiz bir denetleyicisi aygıtları, SCSI ve USB aygıtlarla ilgili olayları yönetir.
 - USB aygıtlar için usbcore, SCSI aygıtlar için scsi_mod çekirdek modülleri bu aygıtlara dönüktür.
 - USB aygıtlar için ayrıca (anakart türüne göre değişen) usb-*hci modülü de yüklenir.



```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ lsmod | grep scsi
scsi_mod          105001  4 sg,sr_mod,sd_mod,libata
bora@pardus-sunucu-32-test:~$ lsmod | grep usb
usbhid            28016  0
hid               50909  1 usbhid
usbcore           99321  4 usbhid,ohci_hcd,ehci_hcd
nls_base          4541   1 usbcore
bora@pardus-sunucu-32-test:~$ lsmod | grep ohci
ohci_hcd          16999  0
usbcore           99321  4 usbhid,ohci_hcd,ehci_hcd
bora@pardus-sunucu-32-test:~$
```

Aygıtlar

- Linux'da DMA kanallarını görmek için /proc/dma dosyası içeriğine bakmak gerekir.
- Ancak bu yolla hangi kanalı hangi sürücünün kullandığını anlamak her zaman mümkün olmaz.
 - PCI aygıtları DMA listesinde gözükmez.
 - Bu durumda dmesg çıktısı içerisinde «dma» sözcüğünü aratmak gerekir.



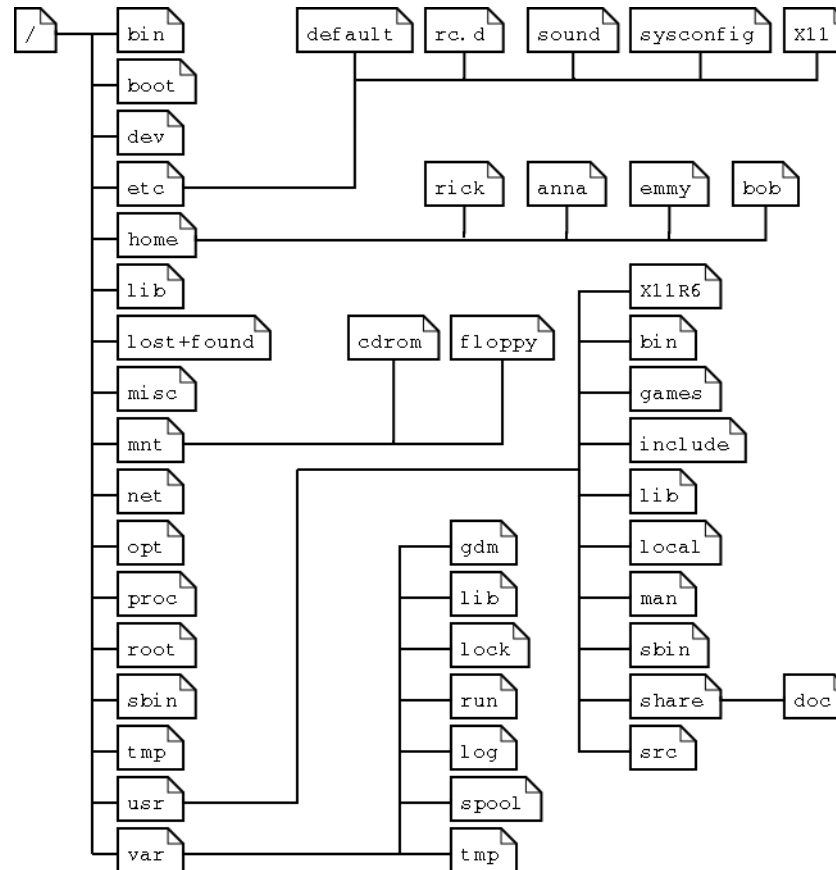
```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ cat /proc/dma
4: cascade
bora@pardus-sunucu-32-test:~$ dmesg | egrep -C 1 'dma'
[ 13.001360] scsi2 : ata_piix
[ 13.006827] ata2: PATA max UDMA/33 cmd 0x1f0 ctl 0x3f6 bmdma 0xd000 irq 14
[ 13.008560] ata3: PATA max UDMA/33 cmd 0x170 ctl 0x376 bmdma 0xd008 irq 15
[ 13.011122] ehci_hcd 0000:00:0b.0: PCI INT A -> Link[LNKKC] -> GSI 10 (level,
low) -> IRQ 10
bora@pardus-sunucu-32-test:~$
```

Dosya Sistemi

Dosya Sistemi

- Linux'da dosya sistemi dediğimiz zaman iki farklı konu akla gelmektedir.
 - Dosya sistemi hiyerarşisi
 - Desteklenen dosya sistemleri
- Linux dosya sistemi hiyerarşisi UNIX geleneğini takip eden ancak güncel ihtiyaçları gözetererek geliştirilmiş bir yapıya sahiptir.
- Bu konuda iki adet standart da mevcuttur.
 - Linux Filesystem Hierarchy Standard (FHS), ABD merkezli Linux Vakfı tarafından 1994'den bu yana sürdürülen ve 2012 içinde en son güncellemesini (taslak olarak) duyuran bir standarttır.
 - Linux Standards Base (LSB), yine Linux Vakfı tarafından Linux ile ilgili bir çok konunun standardize edilmesini hedefleyen bir standarttır ve dosya sistemi hiyerarşisini de gözetermektedir. FHS'in 2012 yılı taslağı LSB'nin güncel haline dayanmaktadır.

Dosya Sistemi



Dosya Sistemi

Dosya Sistemi

Dosya Sistemi

Dosya Sistemi

- Linux dizin hiyerarşisinin bir diğer önemli üstünlüğü erişim izinlerinin hem kullanıcı hem de grup bazlı olarak düzenlenebilmesi ve alt dizinlere yansımalarının seçeneğe bağlı olmasıdır.
 - chown
 - chgrp
 - chmod
- Buna ek olarak ACL olanakları da vardır – ki böylece ince ayar güvenlik mekanizmaları kurulabilsin.

Dosya Sistemi

- Linux için temel dosya erişim hakları modeli, bir dosyaya erişme yetkisi olanların net olarak belirtilmesi ve diğerlerine erişim izni verilmemesi üzerine kurulur.
- Linux işletim sistemi, bu erişim haklarını her bir dosya üzerinde yapılabilecek işlemler ve kullanıcı türüne göre belirler
- Erişim hakları 3 türdür.
 - Okuma hakkı (r), Yazma hakkı (w), Çalıştırma hakkı (x)
 - Her bir dosyaya göreceli olarak kullanıcı türleri de üç türdür.
 - Sahip (u), Grup (g), Diğerleri (o)

Dosya Sistemi

- Var olan dosyalar üzerindeki erişim haklarını ls -l komutu ile listeleyerek görebilirsiniz.
- Peki çıkan listede hangisi erişim haklarını ifade ediyor?
 - drwxr-xr-x 2 root root 4096 Kas 5 01:02 bin
- Böyle bir satırda “drwxr-xr-x” erişim haklarınızı gösterir
 - d harfi bunun bir dizin olduğunu göstermekte
 - Sıradan bir dosya için d harfi yerine – olacaktır.
- Geri kalan harf dizisi 3'erli gruplara bölünür
 - rwx İlk üçlü dosya sahibinin,
 - r-x İkinci üçlü, dosyanın ait olduğu gruptakilerin,
 - r- x Üçüncü üçlü, Diğer kullanıcıların yetkileri

Dosya Sistemi

- Çeşitli sebeplerle dosya erişim izinlerini değiştirmek isteyebilirsiniz.
 - Örneğin dizinlerin (x) yetkisi, o dizini açmak için gereklidir.
- Bunu yapmanın bir yolu, komut satırında `chmod` komutunu kullanmaktır.
- Bu komut 2 şekilde kullanılabilir
 - `chmod [ugoa][+=-][rwxst] dosya_adi`
 - `chmod yetki_izinleri dosya_adi`
 - yetki_izinleri: R (okuma) için 4, w (yazma) için 2, x (çalıştırma) için 1

Dosya Sistemi

- Dosya erişim izinlerini chmod komutu ve sayılar ile değiştirdiğimizde sıklıkla kullanabileceğimiz kombinasyonlar:
 - 755 Sahibi rwx, grubu r-x, diğerleri r-x
 - 750 Sahibi rwx, grubu r-x, diğerleri ---
 - 740 Sahibi rwx, grubu r--, diğerleri ---
 - 550 Sahibi r-x, grubu r-x, diğerleri ---
 - 500 Sahibi r-x, grubu ---, diğerleri ---
 - 400 Sahibi r--, grubu ---, diğerleri ---
- Aynı komutu -R parametresi ile dizinler üzerinde de kullanabiliriz.

Dosya Sistemi

- Eğer yeni oluşturduğunuz dosyaların ya da dizinlerin önceden belirlediğiniz erişim izinleriyle oluşturulmasını istiyorsanız,
 - umask komutu sorunu çözecektir
- Parametre vermeden bu komutu çalıştırırsanız o anki umask değerini görürsünüz.
- Umask komutu ile kaldırmak istediğiniz yetkileri belirtirsiniz.
 - umask yetki_sayıları şeklinde kullanılır.
- umask 022 komutunu verdiğimiz zaman
 - 0: dosya/dizin sahibinden Hiçbir hakkı alma
 - 2: Grup erişim haklarından w(2) yani yazma hakkını al
 - 2: Diğer kişilerden bu dizine yazma hakkını iptal et anlamına gelir

Dosya Sistemi

- Bir dosya ya da dizinin sahibinin değiştirilmesi gerekli olabilir. Bu durumlarda `chown` komutu işinizi görecektir
 - Kendi dosyaları haricindeki dosyalar için, bu komutu sistem yöneticisi yetkileri (süper kullanıcı, super user) olanlar kullanabilir.
 - `chown kullanıcı_ismi dizin/dosya` şeklinde kullanılır
- Örnek vermek gerekirse,

`# chown -R tufan dizin`
- Komutu ile, dizin ve içerisinde bulunan tüm dosya ve dizinlerin hepsi, artık tufan kullanıcıasına atanmıştır.

Dosya Sistemi

- chown komutu, bir dosya yada dizinin hem kullanıcı hem grup ayarlarını değiştirebilir.
- Ancak sadece grup ayarlarını değiştirmek de isteyebilirsiniz.
 - chgrp komutu sadece grup ayarlarını değiştirebilir
- Kullanıcıların kendi dosyaları haricindeki dosyalar için, yine sadece sistem yöneticisi tarafından kullanılır.
 - chgrp grup_ismi dizin/dosya şeklinde kullanılır

Dosya Sistemi

- Sistem yöneticisinin sürekli olarak erişim yetkisi sorunlarını çözmesi yerine, tüm dizinlere veya ayrı programlara bazı özel haklar verilebilir.
- 3 tane özel hak modu bulunur
 - Sticky bit modu
 - Suid biti modu
 - Sgid biti modu

Dosya Sistemi

Sticky Bit

- Eğer bir dosya veya dizin için sticky bit aktif hale getirilirse, o dizin yada dosyanın sahibi silme ve yeniden adlandırma haklarına sahip olacaktır
- /tmp dizinini buna örnek verebiliriz. Bu dizin içerisinde herkes bir dosya oluşturabilir ama bir kullanıcının diğer kullanıcıya ait dosya veya dizini silme hakkı yoktur
- t harfi ile temsil edilir
- `chmod +t dizin_adi` şeklinde kullanılır

Suid Bit

- Suid biti çalıştırılabilir bir dosya üzerine uygulanırsa, dosyayı çalıştıran kişiler programın çalıştığı sürece dosya sahibinin haklarını elde etmiş olur.
- s ile temsil edilir
- `chmod +s dizin_adi/dosya` şeklinde kullanılır
 - Suid biti dikkatli kullanılmalıdır, aksi takdirde güvenlik açıklarına neden olabilir.

Dosya Sistemi

Sgid Bit

- Suid biti ile kullanıcılar için verdiğimiz hakları gruplar için de vermek mümkündür.
- Gruplar için Sgid bitini aktif hale getiririz
- s ile temsil edilir
- `chmod +s grup_ismi dizin_adi/dosya` şeklinde kullanılır

Dosya Sistemi

- ACL'lerin tipik kullanım amaçları şunlardır:
 - Bir kullanıcı adı veya grup ile ifade edilemeyecek kadar karmaşık bir dizi kullanıcının aynı varlığa erişmesini sağlamak
 - Bir varlığa erişme yetkisi olan kullanıcının bu amacın dışında kullanılacak çok fazla yetkisi olması nedeni ile hatalı veya kötü niyetli kullanımdan, yazılım hatalarından veya kötü niyetli kişilerin edimlerinden kaynaklanabilecek zararlara karşı önlem almak.
- Kısacası ACL mekanizması klasik kullanıcı/grup yetkilendirmesinin yeterli olmadığı yerlerde yetkileri artırmak veya azaltmak için kullanılır.
 - ACL konusu ileride daha kapsamlı olarak işlenecektir.

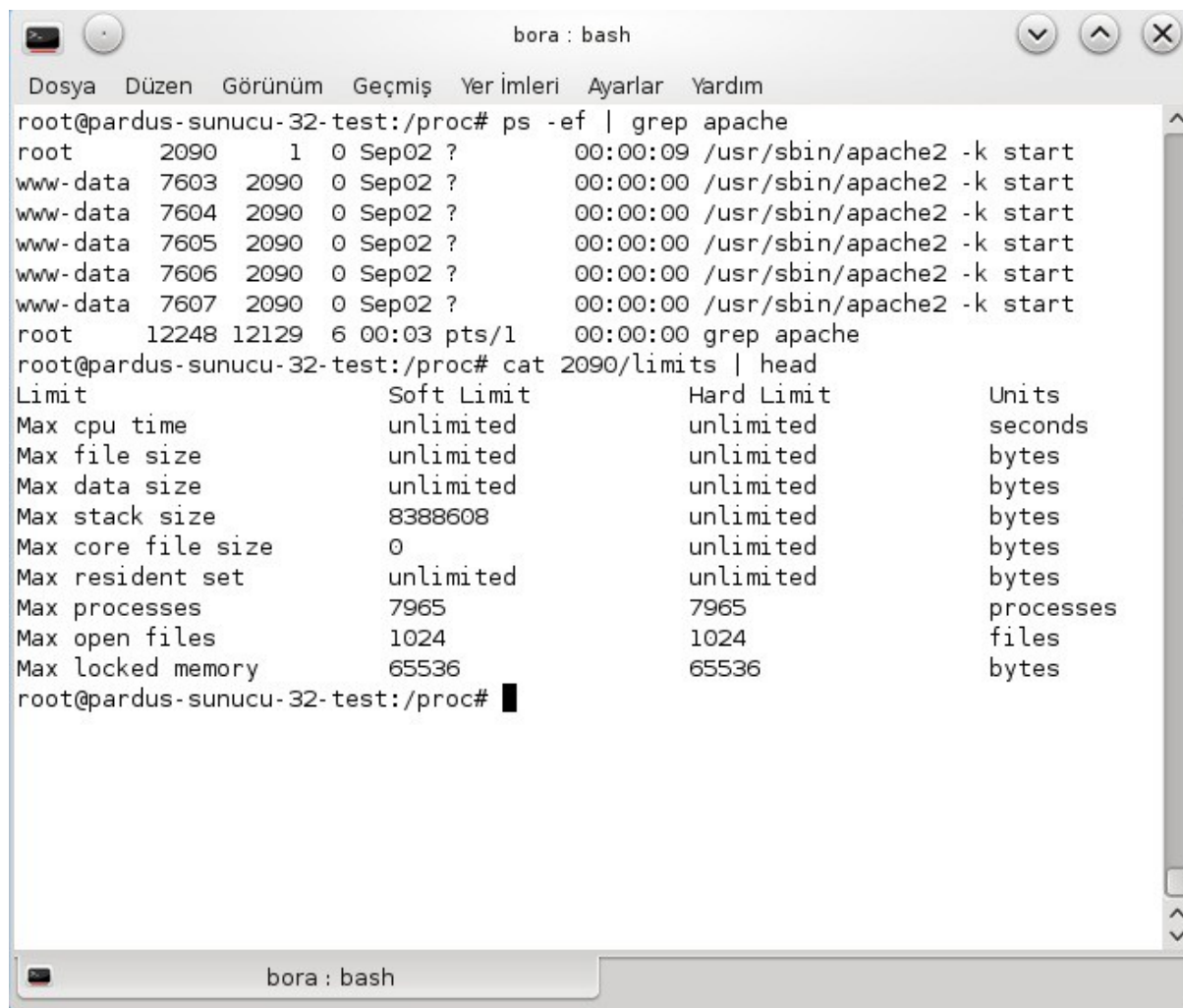
Dosya Sistemi

- ACL'lerin tipik kullanım amaçları şunlardır:
 - Bir kullanıcı adı veya grup ile ifade edilemeyecek kadar karmaşık bir dizi kullanıcının aynı varlığa erişmesini sağlamak
 - Bir varlığa erişme yetkisi olan kullanıcının bu amacın dışında kullanılacak çok fazla yetkisi olması nedeni ile hatalı veya kötü niyetli kullanımdan, yazılım hatalarından veya kötü niyetli kişilerin edimlerinden kaynaklanabilecek zararlara karşı önlem almak.
- Kısacası ACL mekanizması klasik kullanıcı/grup yetkilendirmesinin yeterli olmadığı yerlerde yetkileri artırmak veya azaltmak için kullanılır.
 - ACL konusu ileride daha kapsamlı olarak işlenecektir.

Dosya Sistemi

- Linux için önemli bir sanal dosya sistemi /proc dosya sistemidir.
 - /proc dizini altında numaralandırılmış dizinler ve sistem bilgi dosyaları bulunur.
 - Bu dizin altındaki dosyaların boyutları 0 olur ama cat, vb bir komutla içeriğine baktığımızda içerik olduğunu görürüz.
- Bu dosya sisteminde numaralandırılmış dizinler sürekli olarak ortaya çıkar ve kaybolur.
 - Bunun nedeni her bir dizinin, açılan her bir süreç (İng. Process) için süreç bilgisini vermek üzere otomatik olarak oluşturulmasıdır.
 - Dizinlerin adlarındaki numaralar, süreçlerin süreç tanımlayıcıları (İng. Process ID – pid) olmaktadır.

Dosya Sistemi



```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
root@pardus-sunucu-32-test:/proc# ps -ef | grep apache
root      2090      1  0 Sep02 ?        00:00:09 /usr/sbin/apache2 -k start
www-data  7603    2090  0 Sep02 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  7604    2090  0 Sep02 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  7605    2090  0 Sep02 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  7606    2090  0 Sep02 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  7607    2090  0 Sep02 ?        00:00:00 /usr/sbin/apache2 -k start
root     12248 12129  6 00:03 pts/1    00:00:00 grep apache
root@pardus-sunucu-32-test:/proc# cat 2090/limits | head
Limit                Soft Limit            Hard Limit             Units
Max cpu time         unlimited             unlimited              seconds
Max file size        unlimited             unlimited              bytes
Max data size        unlimited             unlimited              bytes
Max stack size       8388608              unlimited              bytes
Max core file size   0                    unlimited              bytes
Max resident set     unlimited             unlimited              bytes
Max processes        7965                 7965                  processes
Max open files       1024                 1024                  files
Max locked memory    65536                65536                 bytes
root@pardus-sunucu-32-test:/proc#
```

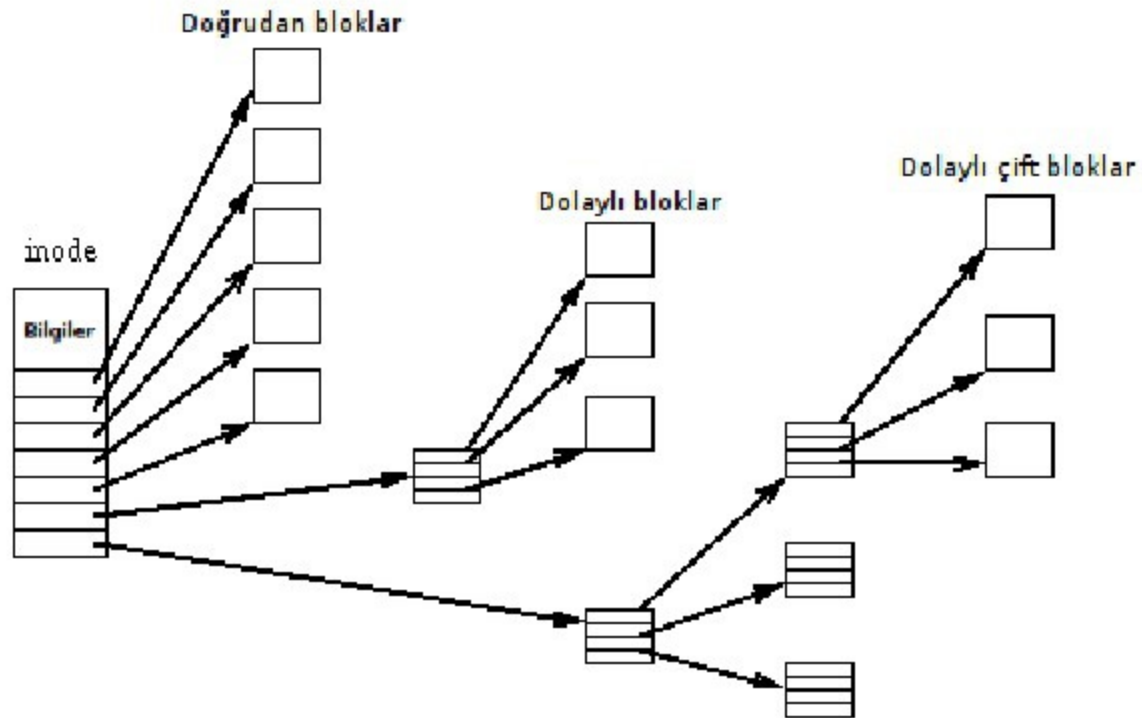
Dosya Sistemi

- /proc dosya sistemindeki bilgi dosyaları sistem ile ilgili olacak bir çok şeyi listeler.
- En sık bakılan bilgiler arasında şunlar yer alır.
 - /proc/cpuinfo – CPU,
 - /proc/meminfo – Bellek,
 - /proc/loadavg – Yük ortalaması,
 - /proc/partitions – Disk bölümleri,
 - /proc/version – Linux sürümü
- Bu dosyaların içeriklerine ayrıca bir man sayfası olarak erişmek mümkündür.
 - Bu amaçla içeriğe «man 5 DOSYAADI» şekline erişebilirsiniz.

Dosya Sistemi

- Linux'da en yaygın kullanılan dosya sistemi ailesi ext dosya sistemi ailesidir.
 - Ext
 - Ext2
 - Ext3
 - Ext4
- Bu dosya sistemi ailesi, ileri özellikleri yanı sıra geriye dönük uyumluluğu ile de öne çıkmaktadır.

Dosya Sistemi



Dosya Sistemi

- Ext3 veya üçüncü uzatılmış dosya sistemi, Linux dağıtımlarında yaygın kullanılan bir dosya sistemidir.
 - Ext3 dosya sistemi, aslında ardılı olduğu ext2 dosya sistemi üzerine ekler içermektedir ve herhangi bir ext3 dosya sistemi, bu ekleri görmezden gelmek yolu ile ext2 olarak da kullanılabilir.
 - Stephen Tweedie tarafından geliştirilen ext3 dosya sisteminin ek özellikleri, güvenilirliği sağlayan günlük tutma becerisi ve istenmeyen bir şekilde sistem kapatılırsa, dosya sistemini kontrol etmesidir.
 - Günlük (journal) tutma becerisi temelde elektrik kesintilerinde dosya kaybetmenizi önler.
- Son 1-2 yıl içinde ext4 dosya sistemi yaygınlaştıkça eski popülerliğini kaybetmektedir.

3. Bölüm

Sistem Yönetim Süreci

- Bu bölümde temel Linux sistem yönetim yaşam çevrimi tartışılacak ve bu kapsamda önemli bazı bilgi ve beceriler kazandırılacaktır.
 - Süreçler ve süreç yönetimi
 - Çekirdek ve kullanıcı uzayı farkları
 - Kabuk ayarları
 - Grafik ortam ayarları
 - Yetkilendirme modeli
 - Sistem yönetim komutları
 - Bash programlama
 - Grep, Sed ve Awk ile programlama

Süreçler ve süreç yönetimi

- Bilgisayarınızda çalışan bir programa süreç adı verilir.
- Bir süreç arka planda çalışan görevlerden (mesela Zemberek, yazım kılavuzu görevi yapar) herhangi birisi olabileceği gibi sistem olayları yakalayıcılarından tamamen gelişmiş uygulamalara (mesela Firefox veya Libre Office) kadar olabilir.
- Tüm süreçler bir ya da daha fazla thread (hafifletilmiş süreç)'lerden oluşur
- Süreçler
 - Hafızada yer kaplar
 - Zaman bazında CPU yu kullanırlar
 - Kesme (İng. Interrupt) alabilir ve üretebilirler

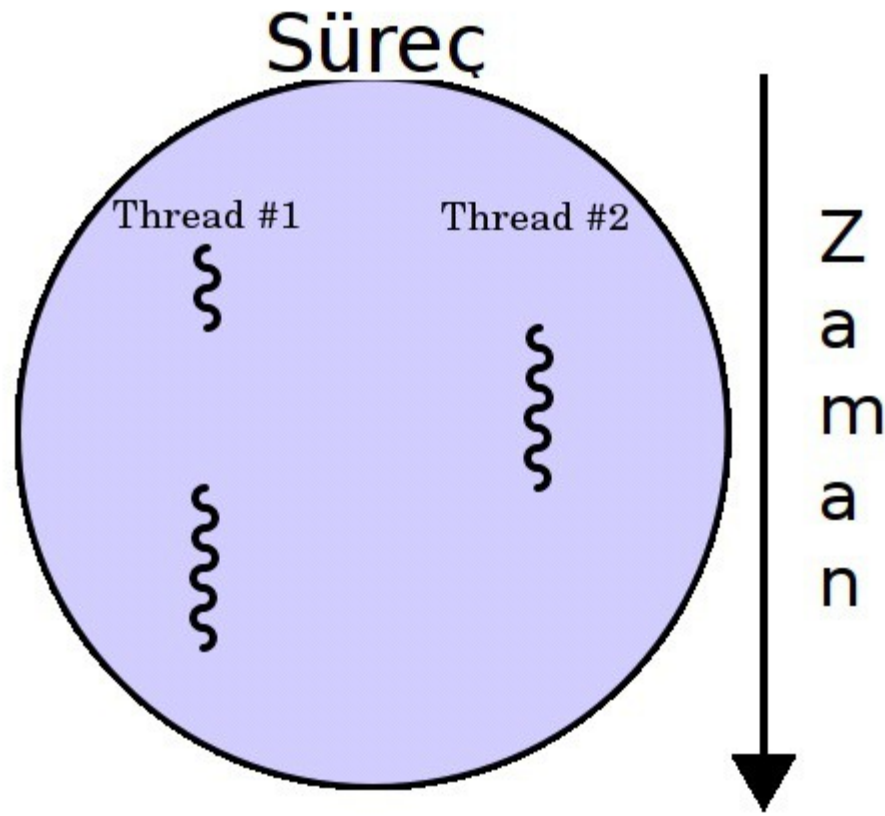
Süreçler ve süreç yönetimi

- Birçok işletim sisteminde ve Pardus'da olduğu gibi, bir çok görev arka planda çalıştığından dolayı, Bilgisayarınızda sizin görüp bildiğiniz programlardan fazla süreçlerin çalışması muhtemeldir.
- Örneğin, bilgisayarınızda sizin bildiğiniz 3 adet program çalışıyor olabilir ama gerçekte 20 tane süreç bilgisayarınızda olabilir.

Süreçler ve süreç yönetimi

- Süreçler ile ilintili bir diğer konu da kanallar yada iş parçacıkları adını verdiğimiz (İng. Thread) mekanizmasıdır.
 - Bir kanalın işletilmesi bir bilgisayar programının bir yada daha fazla göreve bölünerek çalıştırılması anlamına gelir
 - Kanallar süreçlerin içerisinde yer alır
- Çok kanallılık doğru yapıldığı zaman işlemci ve diğer sistem kaynaklarının daha iyi kullanılmasını sağlar.
 - Çok kanallı programlama çoklu görev yapan bir işletim sistemi gerektirir
 - Linux çok kanallı programlama için en iyi ortamlardan birisidir.
 - Red Hat tarafından 2003'de geliştirilen NTPPL kitabığı Linux'da çok yüksek kanal performansı sağlar.

Süreçler ve süreç yönetimi



Süreçler ve süreç yönetimi

- Görevler (İng. Task) ve kanallar birbirine benzer olmakla beraber genelde birbirleriyle karıştırılan terimlerdir. Birçok eski bilgisayar (tek çekirdekli olanlar) bir çalıştırma zamanında en çok bir tane kanalı işleyebilirler. Fakat birçok programı çalıştırıyor ve birçok kullanıcıya eş zamanlı hizmet veriyor gibi çalışırlar.
 - Pardus bu program ve servislere bir sıra verir ve bu sıraya göre programlar işletilir. Bu programların her birine görev adı verilir.
 - Eğer ki bu programlardan bir tanesi G/Ç (girdi/çıkıtı) talebinde bulunursa (mesela bir dosyayı okuma ve yazıcıya yazdırma gibi), o zaman kanal oluşturulur.

Süreçler ve süreç yönetimi

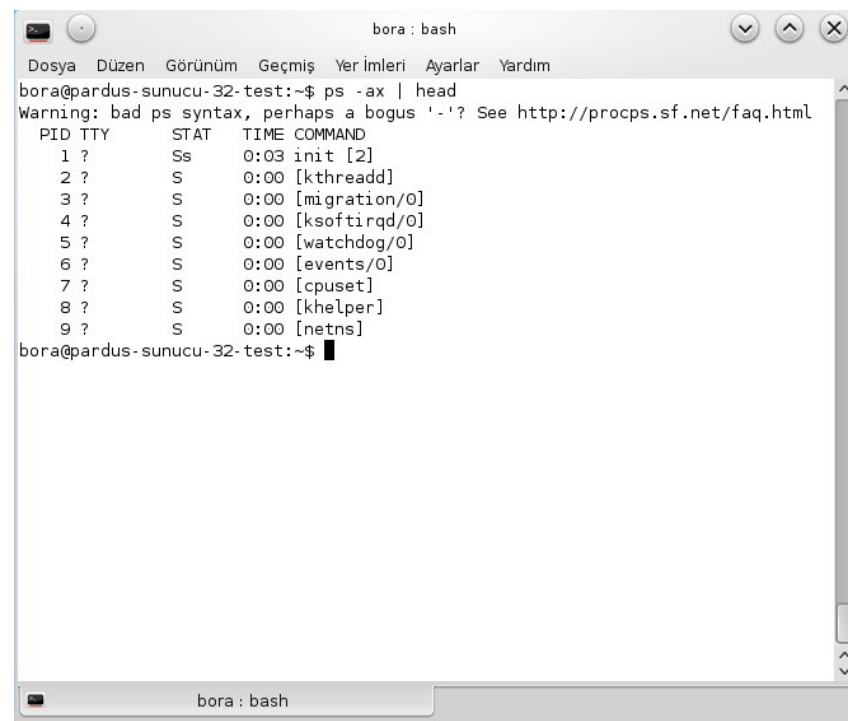
- Süreçler arasındaki ilişkiler şu şekildedir;
 - Sinyaller – Çeşitli şartları işaret etmek için diğer süreçlerden yada çekirdekten özel bir sürece gönderilir.
 - Borular -"]" karakteri ile kabuk tarafından kurulan isim verilmeyen borular bir programın çıktısını diğer programın girdisine yönlendirmeyi sağlar.
 - FIFO - İlk giren ilk çıkar temelleri üzerinde işlem gören isimlendirilmiş borulardır.
 - LIFO – Son giren ilk çıkar temelleri üzerine işlem gören isimlendirilmiş borulardır.
 - Mesaj kuyrukları - bir yada daha fazla diğer süreç tarafından okunabilen mesajları yazmak için bir yada daha fazla sürece izin veren mekanizmadır.
 - Semaforlar - Paylaşılan kaynaklara erişimi kontrol etmek için kullanılan sayaçlardır. Bu sayaçlar bir yada daha fazla sürecin aynı kaynağı aynı zamanda kullanmasını engelleyen kilit mekanizması olarak kullanılır.
 - Paylaşımlı hafıza – birden çok süreç tarafından paylaşılan bir hafıza alanının haritalanmasıdır.

Süreçler ve süreç yönetimi

- Linux'da bir süreçten başka bir süreç oluşturmak için `fork()` adlı C sistem çağrısı kullanılmaktadır.
 - Eğer süreç başka bir süreci oluşturuyorsa “ebeveyn”(parent) süreç, oluşturulan sürece ise “çocuk”(child) süreç ad verilir.
- Süreçlerin aynı anda çalışması ile ilgili 2 tane yaklaşım mevcuttur.
 - Ebeveyn sürecin çocuk süreçlerle aynı anda çalışma imkanı bulunmaktadır.
 - Ebeveyn süreçler çocuk süreçlerin çalışması tamamlanan kadar bekleyebilir.

Süreçler ve süreç yönetimi

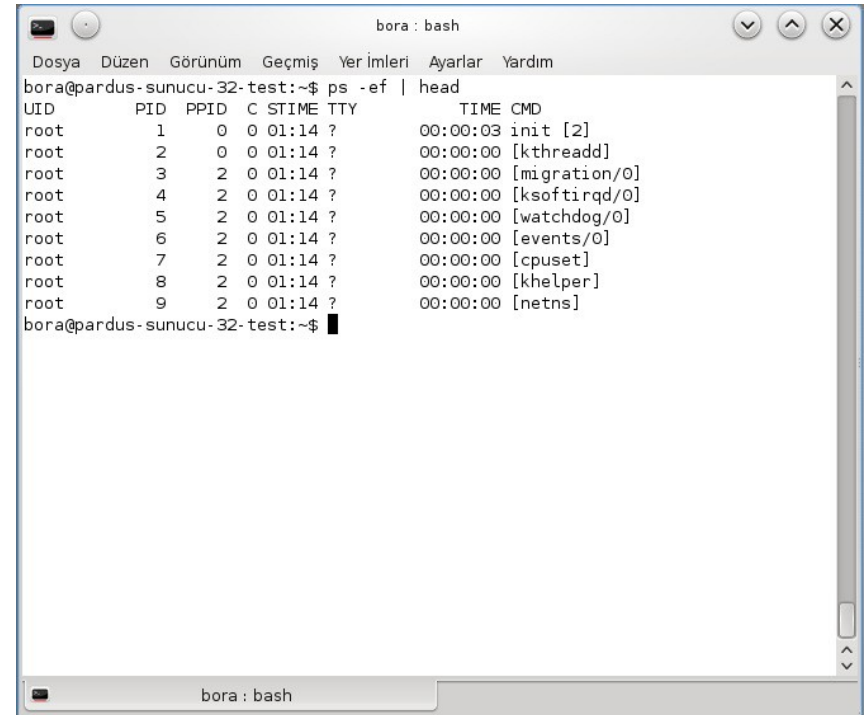
- Süregelen süreçleri izlemek için
 - ps -ax
 - ps -ef



```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ ps -ax | head
Warning: bad ps syntax, perhaps a bogus '- '? See http://procps.sf.net/faq.html
  PID TTY          STAT       TIME COMMAND
    1 ?            Ss        0:03 init [2]
    2 ?            S          0:00 [kthreadd]
    3 ?            S          0:00 [migration/0]
    4 ?            S          0:00 [ksoftirqd/0]
    5 ?            S          0:00 [watchdog/0]
    6 ?            S          0:00 [events/0]
    7 ?            S          0:00 [cpuset]
    8 ?            S          0:00 [khelper]
    9 ?            S          0:00 [netns]
bora@pardus-sunucu-32-test:~$
```

Süreçler ve süreç yönetimi

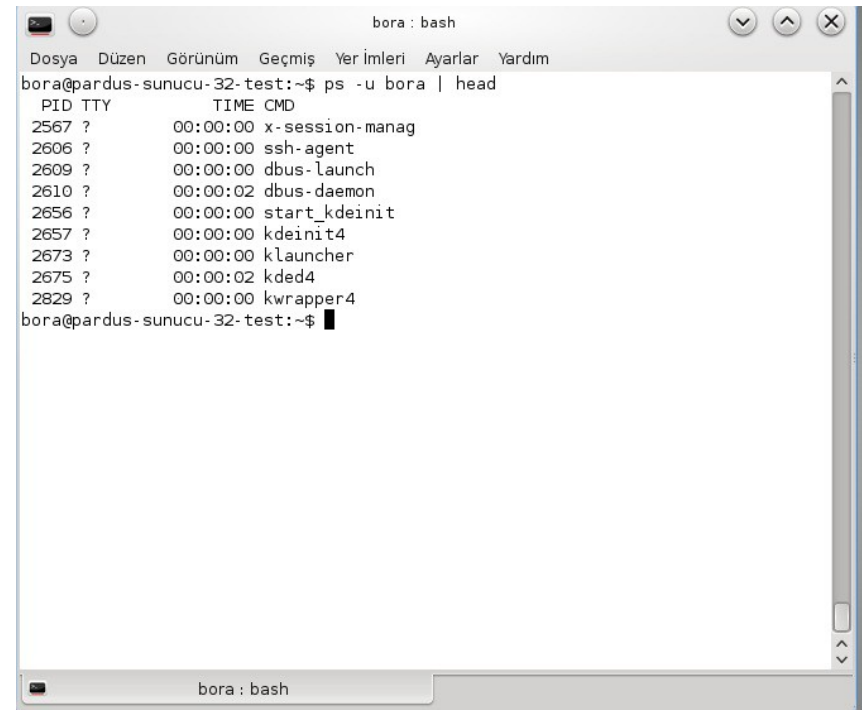
- Süregelen süreçleri izlemek için
 - ps -ax
 - ps -ef



```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ ps -ef | head
UID          PID  PPID  C  STIME TTY          TIME CMD
root          1    0  0  01:14 ?           00:00:03 init [2]
root          2    0  0  01:14 ?           00:00:00 [kthreadd]
root          3    2  0  01:14 ?           00:00:00 [migration/0]
root          4    2  0  01:14 ?           00:00:00 [ksoftirqd/0]
root          5    2  0  01:14 ?           00:00:00 [watchdog/0]
root          6    2  0  01:14 ?           00:00:00 [events/0]
root          7    2  0  01:14 ?           00:00:00 [cpuset]
root          8    2  0  01:14 ?           00:00:00 [khelper]
root          9    2  0  01:14 ?           00:00:00 [netns]
bora@pardus-sunucu-32-test:~$
```


Süreçler ve süreç yönetimi

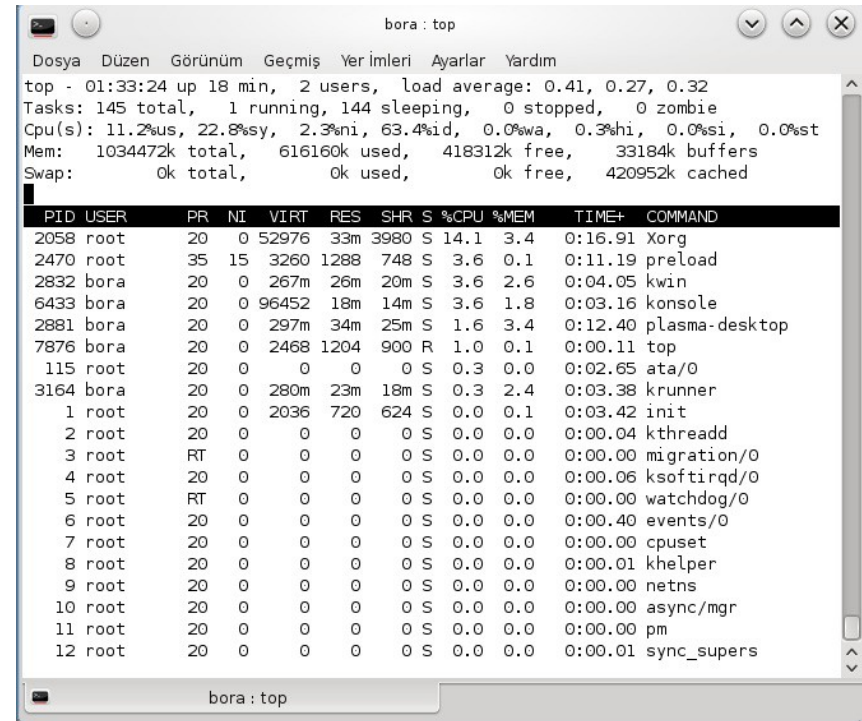
- Belirli bir kullanıcının süreçlerini tespit etmek için ise
 - `ps -u KULLANICI`



```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer imleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ ps -u bora | head
  PID TTY          TIME CMD
 2567 ?        00:00:00 x-session-manag
 2606 ?        00:00:00 ssh-agent
 2609 ?        00:00:00 dbus-launch
 2610 ?        00:00:02 dbus-daemon
 2656 ?        00:00:00 start_kdeinit
 2657 ?        00:00:00 kdeinit4
 2673 ?        00:00:00 klauncher
 2675 ?        00:00:02 kded4
 2829 ?        00:00:00 kwrapper4
bora@pardus-sunucu-32-test:~$
```

Süreçler ve süreç yönetimi

- Sık kullanılan süreç yönetim komutları
 - top
 - pstree
- Bu komutlardan top süreçlerin çalışma zamanında tükettikleri temel kaynakları izlemeye yarar.
- pstree komutu ise sistemde bulunan süreçleri ve ona bağlı çocuk süreçlerin listesini ağaç yapısında göstermeye yarar



```
bora : top
Dosya Düzen Görünüm Geçmiş Yer imleri Ayarlar Yardım
top - 01:33:24 up 18 min,  2 users,  load average: 0.41, 0.27, 0.32
Tasks: 145 total,  1 running, 144 sleeping,  0 stopped,  0 zombie
Cpu(s): 11.2%us, 22.8%sy,  2.3%ni, 63.4%id,  0.0%wa,  0.3%hi,  0.0%si,  0.0%st
Mem:  1034472k total,  616160k used,  418312k free,  33184k buffers
Swap:      0k total,      0k used,      0k free,  420952k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2058 root        20   0 52976  33m 3980 S 14.1   3.4   0:16.91 Xorg
 2470 root        35  15  3260 1288  748 S   3.6   0.1   0:11.19 preload
 2832 bora        20   0  267m  26m  20m S   3.6   2.6   0:04.05 kwin
 6433 bora        20   0 96452  18m  14m S   3.6   1.8   0:03.16 konsole
 2881 bora        20   0  297m  34m  25m S   1.6   3.4   0:12.40 plasma-desktop
 7876 bora        20   0  2468 1204  900 R   1.0   0.1   0:00.11 top
  115 root        20   0     0     0     0 S   0.3   0.0   0:02.65 ata/0
3164 bora        20   0  280m  23m  18m S   0.3   2.4   0:03.38 krunner
    1 root        20   0  2036  720  624 S   0.0   0.1   0:03.42 init
    2 root        20   0     0     0     0 S   0.0   0.0   0:00.04 kthreadd
    3 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 migration/0
    4 root        20   0     0     0     0 S   0.0   0.0   0:00.06 ksoftirqd/0
    5 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 watchdog/0
    6 root        20   0     0     0     0 S   0.0   0.0   0:00.40 events/0
    7 root        20   0     0     0     0 S   0.0   0.0   0:00.00 cpuset
    8 root        20   0     0     0     0 S   0.0   0.0   0:00.01 khelper
    9 root        20   0     0     0     0 S   0.0   0.0   0:00.00 netns
  10 root        20   0     0     0     0 S   0.0   0.0   0:00.00 async/mgr
  11 root        20   0     0     0     0 S   0.0   0.0   0:00.00 pm
  12 root        20   0     0     0     0 S   0.0   0.0   0:00.01 sync_supers
```

Süreçler ve süreç yönetimi

- Süreç öldürmek genellikle başlatılan program kontrolden çıkınca ya da beklemediğiniz bir biçimde çalışıyorsa başvurulan yöntemdir.
 - İlk önce denenecek olan yöntem Ctrl+C tuş kombinasyonu o da olmazsa Ctrl+D tuş kombinasyonlarını denemektir. Elbette bunun için Çalıştırılan programın ekranı açık olması ve o sürecin sahibi olmanız şart.
 - Eğer bahsedildiği gibi süreci öldürmeyi başaramıyorsanız sürecin numarasını öğrenip sürece öl mesajı göndermeniz gerekmektedir.

Süreçler ve süreç yönetimi

- Süreçleri yeniden başlatmak için «kill –secenek süreç_numarası» komutunu aşağıdaki parametreler ile çağırmak yeterli olacaktır
 - SIGHUP (1) : Yeniden başlatma
 - SIGCONT (18): Süreci devam ettirme
- SIGHUP komutu zombi duruma geçmiş cevap vermeyen ya da başka sebeplerden dolayı süreci yeniden başlatmak istediğimiz zaman kullanılır. Süreç yeniden başlatıldığı için sürece bağlı verilerde kaybolur.
- SIGCONT durdurulan süreci devam ettirdiği için sürece bağlı veriler kaybolmaz.SIGHUP ve SIGCONT diğer sinyallerin de olduğu gibi signal.h dosyası içerisinde tutulur.

Süreçler ve süreç yönetimi

- Süreçleri dondurmak için «kill –secenek süreç_numarası» komutunu aşağıdaki parametreler ile çağırmak yeterli olacaktır:
 - SIGSTOP(17,19,23) : Süreci durdurma

Çekirdek ve kullanıcı uzayı farkları

Çekirdek Uzayı

- Katı biçimde çekirdek ve uzantılarının çalışması için ayrılan bellektir.
- Sayfalama yapılmaz, kodlar mutlak adres uzaylarında çalışır.
- Kullanıcı süreçlerinin sadece bazı API fonksiyonlarını çağırarak erişebildiği bir alandır.

Kullanıcı Uzayı

- Katı biçimde kullanıcı uygulamalarının kullanıcı yetkileri ile çalıştırılması için ayrılan bellektir.
- Sayfalama yapılır.

Kabuk ayarları

- Bir Unix terimi olan kabuk (shell), işletim sistemi ile olan etkileşimli kullanıcı arayüzüdür.
 - Kabuk, kullanıcının girdiği komutları anlayıp çalıştıran programlama katmanıdır.
 - Bazı sistemlerde kabuk, komut yorumlayıcısı (command interpreter) veya komut satırı (command line) olarak da adlandırılır.
- İşletim sisteminin daha üst bir katmanında bulunan kabuk, sıradan kullanıcılar tarafından işletim sisteminin en iç kısmında bulunan çekirdek servisleriyle karıştırılabilir.
 - Kabukta çalıştıracığınız bazı komutlar çekirdek servislerini kullanabilir ancak kabuk çekirdek seviyesinde çalışmaz.

Kabuk ayarları

- Kabuk
 - Bir programdır
 - Bir kullanıcı arayüzüdür
 - Bir yorumlayıcıdır
 - Bir programlama dilidir
- Pardus ile gelen standart kabuk Bash kabuğudur.



```
#!/bin/bash
```


Kabuk ayarları

- Bash'de komut satırı düzenleme, farklı araçlarda da tarafından kullanılan Readline kütüphanesi tarafından sağlanır. Dolayısı ile aşağıdaki kısayol tuşlarını bir çok yerde göreceksiniz.
 - Ctrl+k, imleçten sonraki herşeyi siler
 - Ctrl+b, bir karakter geri gider
 - Ctrl+f, bir karakter ileri gider
 - DEL, imlecin soluna doğru karakterleri siler
 - Ctrl+d, imlecin altındaki karakteri siler
 - Ctrl+a, satır başına gidilir
 - Ctrl+e, satır sonuna gidilir

Kabuk ayarları

- Pardus'da sisteme giriş yaptığınız zaman, aynı zamanda bir kabuk oturumu da başlatmış olursunuz.
- Bu oturum süresince kullandığımız bazı programlar, kendi çalışmalarını düzenleyen çevre değişkenlerine ihtiyaç duyar.
- Bash kabuğunda bir çevre değişkeni oluşturmak için
 - `değişken_ismi=değer` Ya da
 - `export değişken_ismi=değer`
- Kullanacağımız komutların, program dosyalarının hangi dizinde bulunacağını belirtmek istediğimizde
 - `PATH=/etc:/media:/dev:/usr:` şeklinde kullanabiliriz

Kabuk ayarları

- Bash kabuğunda bazı değişkenler özel karakterler ile temsil edilip bazı özel anlamları vardır:
 - \$1...\$n: komuta verilen argüman sayısına göre değişir. Örneğin \$1 komuta ilk girilen argümanı temsil eder
 - \$\$: çalışmakta olan sürecin numarasını verir
 - \$?: çalıştırılan son komutun çıkış kodunu verir
 - \$0: kabuğun ismini verir
 - !\$: girilen son komutun son argümanını alır
 - !!: girilen son komutu alır
- Yukarıdaki karakterler bir değişken adıdır. Bu değişken adları sadece okunabilir, yazılamazlar. Okumak için echo komutunu kullanabiliriz.

Kabuk ayarları

- Önemli bash dosyaları:
 - .bashrc
 - .bash_logout
 - .bash_history
- Yukarı ok tuşu ile önceden girdiğimiz komutlara ulaşabiliriz.
- Aşağı ok tuşu ise önceden girilen komutlara diğer yönde erişmek için kullanılır.
- Ctrl+R tuş kombinasyonu ile önceden girilen komutları arayabilirsiniz

Kabuk ayarları

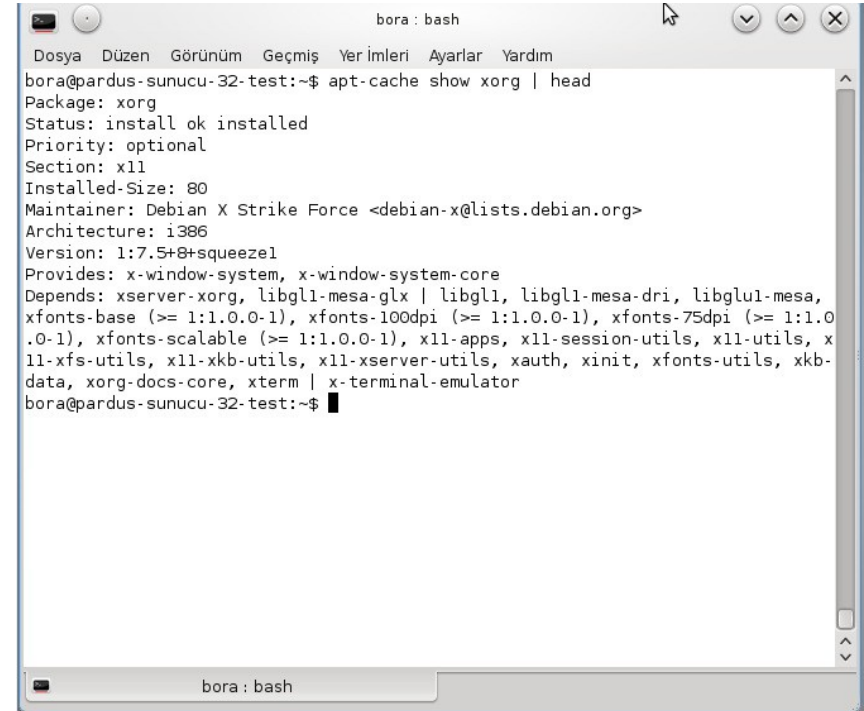
- Çalıştırmak için girdiğiniz program isminin bir kısmını unuttuysanız, hatırladığınız kadarını girin ve sekme (tab) tuşuna basın.
- Eğer komutun adını biliyorsanız ama parametre listesini bilmiyorsanız, sekme argüman girileceği zaman argümanı bekler.
 - Aynı sistem dosya ve dizinlerde de geçerlidir. Sekme ile uzun dosya adlarını tamamlayabilirsiniz.
 - Bu içinde boşluk karakteri bulunan dosyalarda özellikle işe yarar. Normalde bu dosyaların adlarını girerken çift tırnak veya ters bölme karakterlerini bolca kullanmanız gerekir.
 - Aynı şekilde alt alta bir çok dizin olduğu zaman konumları bulmak için de yararlı bir özelliktir.

Kabuk ayarları

- Grafik arayüze sahip olunmadığı yada bir neden ile kullanılmadığı durumlarda (örneğin uzaktan bağlantı) nano sizin için oldukça kullanışlı bir araç olacaktır.
- Bir düz metin editörü olan Nano sade ancak güçlü bir araçtır. Grafik arayüzde çalışan pencere tabanlı (Kate ve Kwrite gibi) editörler kadar güçlü olmasa bile, bir fareye ihtiyaç duyulmadan birçok işlemi yerine getirebilirsiniz.
- Çoğu nano komutları Ctrl ve diğer tuşların kombinasyonu kullanılarak çağrılır.
- Nano (geliştirilmiş pico) olmadığı durumlarda isterseniz pico ve vim editörlerini de deneyebilirsiniz
- Varolan dosyayı nano ile açmak için
 - nano dosya_adi
 - komutunu verebilirsiniz.

Grafik ortam ayarları

- Hangi pencere yöneticisinin kullanıldığından bağımsız olarak X.org alt yapısı kullanmaksızın grafik ortamda çalışmak mümkün değildir.
- Bu nedenle X.org'un ayarları KDE veya Gnome ayarlarına göre daha düşük seviyede ayarlar içermektedir.
- Pardus X.org'un 7.5 sürümünü kullanmaktadır.



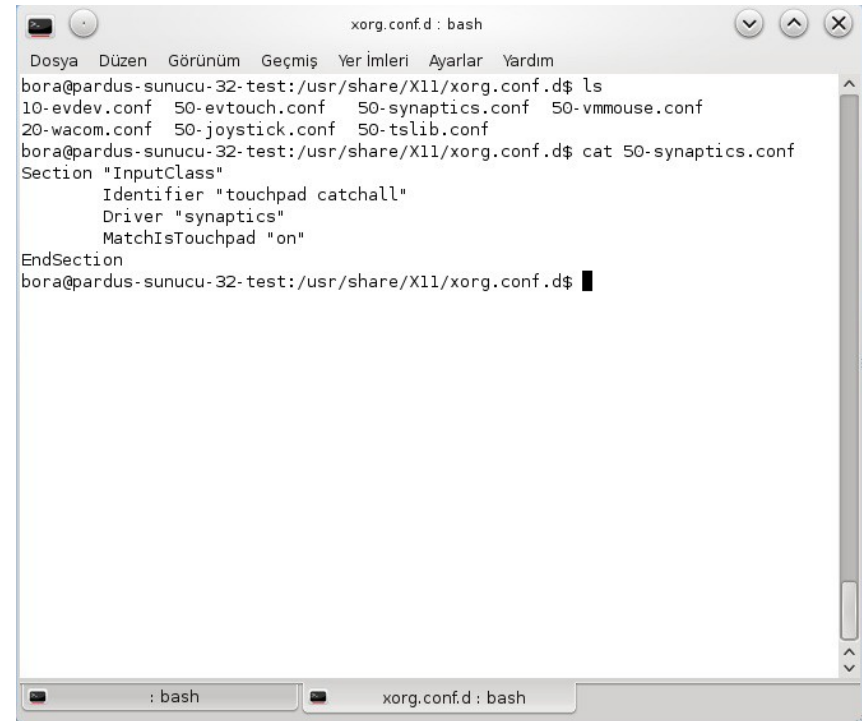
```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
bora@pardus-sunucu-32-test:~$ apt-cache show xorg | head
Package: xorg
Status: install ok installed
Priority: optional
Section: x11
Installed-Size: 80
Maintainer: Debian X Strike Force <debian-x@lists.debian.org>
Architecture: i386
Version: 1:7.5+8+squeeze1
Provides: x-window-system, x-window-system-core
Depends: xserver-xorg, libgl1-mesa-glx | libgl1, libgl1-mesa-dri, libgl1-mesa,
xfonts-base (>= 1:1.0.0-1), xfonts-100dpi (>= 1:1.0.0-1), xfonts-75dpi (>= 1:1.0
.0-1), xfonts-scalable (>= 1:1.0.0-1), x11-apps, x11-session-utils, x11-utils, x
11-xfs-utils, x11-xkb-utils, x11-xserver-utils, xauth, xinit, xfonts-utils, xkb-
data, xorg-docs-core, xterm | x-terminal-emulator
bora@pardus-sunucu-32-test:~$
```

Grafik ortam ayarları

- X.org'un ana ayar dosyası `/etc/X11/xorg.conf` adresindedir.
- Bu dosya silinecek olursa, sistemi tek kullanıcı kipinde açıp arkasından aşağıdaki komutları çalıştırmak gerekir:
 - `cd /etc/X11/`
 - `Xorg -configure`
- Grafik arayüzün açılmaması ve bir konsol girdisi ile karşılaşılması durumunda, grafik arayüzü çalıştırmak için alternatif bir kaç komut bulunmaktadır.
 - `startx` (Evrensel)
 - `service kdm start` (KDE)
 - `Service gdm start` (Gnome)

Grafik ortam ayarları

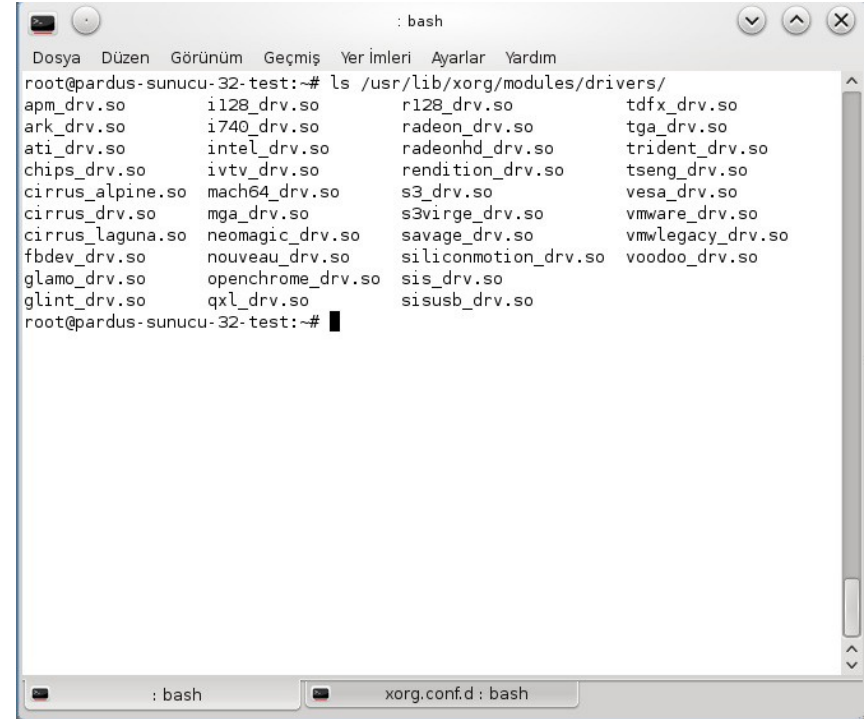
- X.org için aygıt ayarlarına (girdi ve çıktı) dönük ayarlardaki hatalar tipik çöküş veya kilitlenme nedenleridir.
 - Belli başlı ayarlar zaten KDE veya Gnome'un yazılımları tarafından doğru biçimde yapılır ancak bazen elle düzenleme gerekebilir.
 - Hatalı ayarlar X.org'un çökmesine neden olabileceği için öncelikle eski ayarları yedeklemekte fayda vardır.



```
xorg.conf.d : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
bora@pardus-sunucu-32-test:/usr/share/X11/xorg.conf.d$ ls
10-evdev.conf 50-evtouch.conf 50-synaptics.conf 50-vmouse.conf
20-wacom.conf 50-joystick.conf 50-tslib.conf
bora@pardus-sunucu-32-test:/usr/share/X11/xorg.conf.d$ cat 50-synaptics.conf
Section "InputClass"
    Identifier "touchpad catchall"
    Driver "synaptics"
    MatchIsTouchpad "on"
EndSection
bora@pardus-sunucu-32-test:/usr/share/X11/xorg.conf.d$
```

Grafik ortam ayarları

- X.org için en önemli çıktı aygıtı ekrandır.
- Ekran kartı sürücüsü özünde X.org için gereklidir.
 - Bu nedenle ekran kartı sürücüleri `/usr/lib/xorg/modules/drivers` dizini altında yer alır.



```
root@pardus-sunucu-32-test:~# ls /usr/lib/xorg/modules/drivers/
apm_drv.so      i128_drv.so    r128_drv.so    tdfx_drv.so
ark_drv.so      i740_drv.so    radeon_drv.so  tga_drv.so
ati_drv.so      intel_drv.so    radeonhd_drv.so trident_drv.so
chips_drv.so    ivtv_drv.so    rendition_drv.so tseng_drv.so
cirrus_alpine.so mach64_drv.so   s3_drv.so      vesa_drv.so
cirrus_drv.so   mga_drv.so     s3virge_drv.so vmware_drv.so
cirrus_laguna.so neomagic_drv.so savage_drv.so   vmlegacy_drv.so
fbdev_drv.so   nouveau_drv.so siliconmotion_drv.so voodoo_drv.so
glamo_drv.so   openchrome_drv.so sis_drv.so
glint_drv.so   qxl_drv.so     sisusb_drv.so
root@pardus-sunucu-32-test:~#
```

Kullanıcı ve Grup Yönetimi

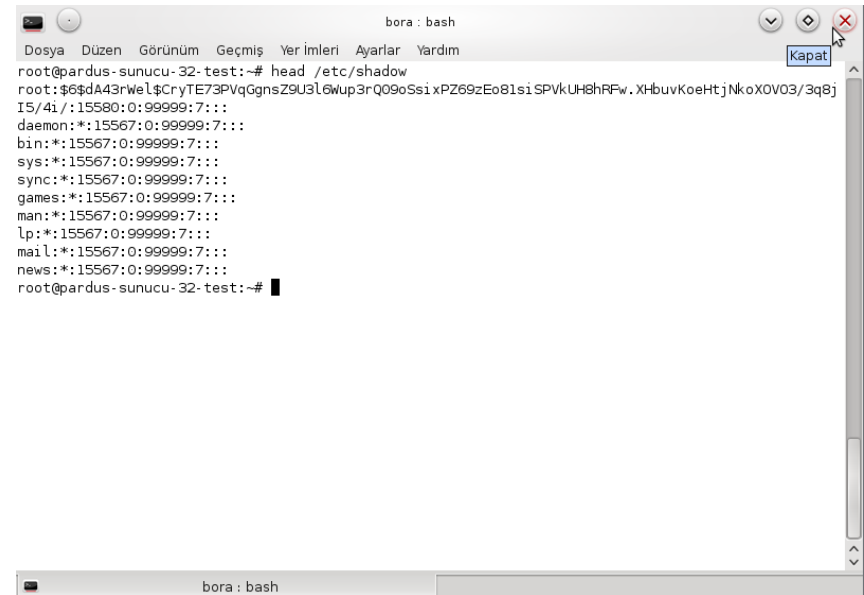
- Yeni bir kullanıcı hesabı açmak yöneticilerinin sorumluluklarının sadece bir kısmıdır.
 - Hesap açmak, kullanıcı ve gerekirse grup yaratmak anlamına gelir
 - Hesap açarken kullanıcıya mutlaka bir parola verilmelidir
 - Bazı kullanıcıların ev dizinleri olmayabilir veya ev dizini diğerlerinden farklı yerlerde olabilir
- Buna ek olarak, sistem kaynaklarını belirleme ve sınırlandırma gibi eylemleri yerine getirmek de görevler de kullanıcı hesapları ile ilintilidir.

Kullanıcı ve Grup Yönetimi

- Linux'da kullanıcı parolaları güvenli olarak saklanır.
- Adından anlaşıldığının aksine /etc/passwd dosyası parolalarla ilgili hiçbir bilgi içermez.
- Parolaların kriptografik özetleri, ise /etc/shadow dosyasında saklanır.
- Bu dosyalardan /etc/passwd dosyasına erişim açıkken, /etc/shadow dosyasına erişim için root kullanıcı izinlerine ihtiyaç vardır.

Kullanıcı ve Grup Yönetimi

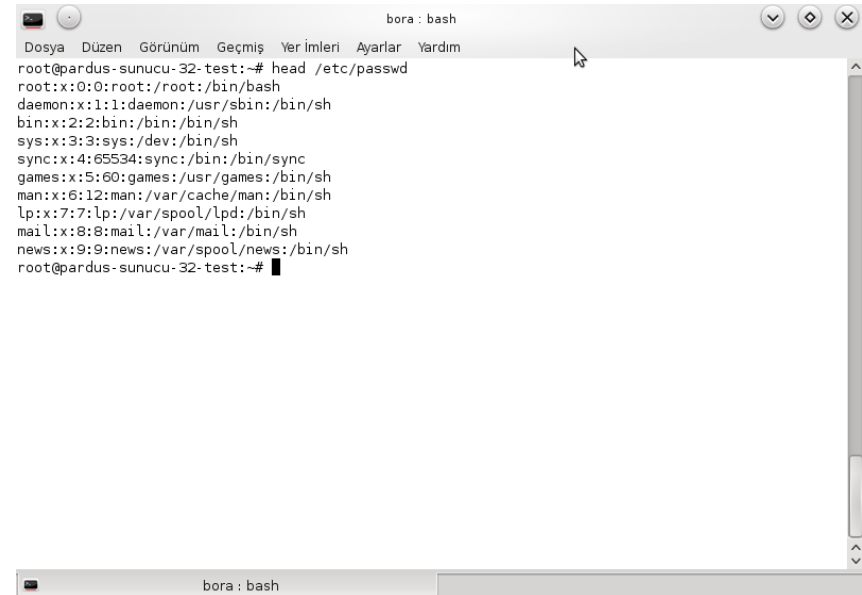
- /etc/shadow dosyasında “:” karakteri ayraç karakteri olarak kullanılır.
- Her bir “:” arasındaki bölgeler aşağıdaki anlama gelir:
 - Kullanıcı ismi
 - Parolanın özeti
 - Son parola değişim tarihi
 - Asgari parola değiştirme periyodu
 - En uzun parola geçerlilik süresi
 - Parolanın zaman aşımı tarihi
 - Parolanın etkisizleştirilmesinden sonraki toplam gün sayısı
 - Bu girişin artık kullanılmadığı



```
bora : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
root@pardus-sunucu-32-test:~# head /etc/shadow
root:$6$dA43rWl$CryTE73PVqGgnsZ9U3L6Wup3rQ09oSs1xPZ69zEo81siSPVkuH8hRFw.XHbuvKoeHtjNkoXOV03/3q8j
IS/4i/:15580:0:99999:7:::
daemon*:15567:0:99999:7:::
bin*:15567:0:99999:7:::
sys*:15567:0:99999:7:::
sync*:15567:0:99999:7:::
games*:15567:0:99999:7:::
man*:15567:0:99999:7:::
lp*:15567:0:99999:7:::
mail*:15567:0:99999:7:::
news*:15567:0:99999:7:::
root@pardus-sunucu-32-test:~#
```

Kullanıcı ve Grup Yönetimi

- /etc/passwd dosyası kullanıcı sisteme giriş yaptığı zaman kullanılır
 - x karakteri parolanın özetinin /etc/shadow dosyası içerisinde tutulduğunu gösterir
 - Kullanıcı numarası bu bölümde tutulur. 0 kullanıcı numarası root kullanıcıasına ait olup, 1-99 arası numaralar ise ayrılmış numaralardır. 100-999 arası numaralar sistem hesap ve hesapları için ayırttırılmıştır.
 - Grup numarası
 - Kullanıcı numarası bilgisi
 - Ev dizini
 - Komut/kabuk



```
bora: bash
Dosya Düzen Görünüm Geçmiş Yerimleri Ayarlar Yardım
root@pardus-sunucu-32-test:~# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
root@pardus-sunucu-32-test:~#
```

Kullanıcı ve Grup Yönetimi

- Pardus'da ve grafik arayüz olanakları sunan tüm Linux dağıtımlarında kullanıcı ve gruplarla ilgili işlemleri yapmak için temel bir «kullanıcı yöneticisi» aracı bulunur.
- Bu araçlarla işlem yapılırken sistem yöneticisi hakları gerekli olacaktır.



Kullanıcı ve Grup Yönetimi

- Linux kullanırken bir kullanıcı hesabı açmanın bir çok yolu vardır; çünkü hesap açma diye adlandırılan eylem, temelde /etc/passwd ve /etc/shadow dosyalarına satır eklemekten ve kullanıcı için bir ev dizini oluşturmaktan başka bir şey değildir.
 - Eğer tam olarak ne yazacağınızı biliyorsanız, bu dosyaları elinizle bile düzenleyebilirsiniz ancak bu zahmetli bir iştir.
 - Bu nedenle komut satırında bu iş için geliştirilmiş bir dizi komut kullanılır.

Kullanıcı ve Grup Yönetimi

Kullanıcı Ekleme

- Tüm Linux dağıtımlarında olduğu gibi Pardus'da da komut satırındaki araçlar ile kullanıcı hesabı işlemleri yapılabilir.
 - Burada kullanıcı işlemleri için iki ayrı komut bulunmaktadır: `adduser` ve `useradd`.
 - Komut satırı uygulamalarında daha çok `adduser` komutu kullanılır.

Kullanıcı Ekleme

- Eklenen bir kullanıcının en temel parametreleri şunlardır.
 - Kullanıcı adı
 - Gerçek adı
 - Grubu
 - Kullanıcı numarası
 - Ev dizini
 - Varsayılan kabuğu
- Kullanıcı parolasının belirlenmesi ikincil bir işlemidir ve daha sonra da yapılabilir.
 - Bazı kullanıcıların asla bir parolası olmayabilir.

Kullanıcı ve Grup Yönetimi

Parola Oluşturmak

- Komut satırında hızlı biçimde yaratılan bir kullanıcının parolası oluşturulmamıştır. Dolayısı ile sisteme giriş yapamaz.
- Parola oluşturmak için, parola değiştirmekte de kullandığımız passwd komutunu kullanırız.

Hesap Ayarlarının Yapılması

- Kullanıcı parametrelerini ayarlamak için sıklıkla kullanılan bir diğer komut da usermod komutudur.
- usermod parametre[ler]
kullanıcı_ismi
 - [-c açık_adı]
 - [-d kişisel_dizin]
 - [-s kabuk]
 - [-p şifre]
 - [-u kullanıcı_sayısal_kodu]
 - [-G grup1[,grup2[, ...]]]

Kullanıcı ve Grup Yönetimi

Hesap Zaman Aşımı

- `usermod -e 2012-12-29 kullanıcı`
 - Komutu kullanıcının parolasının 29 aralık 2012'ye kadar geçerli olduğunu belirtir. Yani zaman aşımı ayarı yapılmıştır.
 - Sistem yöneticisi yeni bir parola verene kadar kullanıcı sisteme bağlanamayacaktır.
 - Kullanıcılar sisteme tekrar girdikleri zaman, zaman aşımı yaklaşıyorsa, sistem kullanıcıyı uarmaya başlar. Ancak kabuk kullanmayan kullanıcıların bu mesajı görmesi mümkün değildir.

Hesabın Kilitlenmesi

- Bir hesaba erişimi geçici olarak kapatmak isteyebilirsiniz. Bu amaçla
 - «`usermod -L kullanıcı`» komutu ile hesabı geçici olarak devre dışı bırakıp,
 - «`usermod -U kullanıcı`» komutu ile ise hesabı tekrar aktif hale getirebilirsiniz.
 - L parametresi İngilizce kilitlemek anlamına gelen “lock”, U parametresi ise kilidi açmak anlamında gelen “unlock” sözcüğünün kısaltmasıdır.

Kullanıcı ve Grup Yönetimi

Kullanıcı Hesabının Silinmesi

- Komut satırında kullanıcı hesabı silmek için `userdel` komutu kullanılır.
 - «`userdel [-r] kullanıcı_ismi`» komutunu kullanarak kullanıcıyı sistemden kaldırabilirsiniz.
 - Kullanıcının kişisel dizinini de silmek isterseniz `-r` yi kullanmanız gerekmez
- Ayrıca `/etc/shadow` ve `/etc/passwd` dosyalarında ilgili satırları silerek de kullanıcıyı silmiş olursunuz.

İskelet (Skel) Dizini

- `/etc/skel` dizini `useradd` ile oluşturulan kullanıcının ev dizinine otomatik olarak kopyalanan dosya ve dizinleri içerir.
- Pardus başlatıldığında, kullanıcı konfigürasyon dosyaları `/etc/skel` içerisinde yerleştirilir.
 - Tipik olarak, `.bash_profile`, `dircolors`, `.inputrc` ve `.vimrc` gibi gizli dosyaları içerebilir.
 - `/etc/skel` dizininin içeriği `ls -a` komutu ile gösterilebilir.

Kullanıcı ve Grup Yönetimi

- Kullanıcıları gruplandırarak erişim yetkilerini düzenlemek gerçekten yararlı bir yöntemdir.
 - Mesela bir projede çalışan tüm kullanıcıları aynı gruba yerleştirip yetkilerini düzenlemek ve proje ile ilgili dosya ve dizinlerin erişim yetkilerini düzenlemek kullanışlı olabilir.
 - Aynı zamanda bir kullanıcının birden fazla grupta yer alması mümkündür.
- Kullanıcı gruplarına dair kayıtlar “/etc/group” dosyasında saklanır.

Kullanıcı ve Grup Yönetimi

- Sisteme yeni bir grup eklemek istediğimiz zaman /etc/group dosyasına bir satır eklemiş oluruz.
 - Bunu yapmak için komut satırında «groupadd [-g grup_numarası] grup_ismi» komutu ile de grup ekleyebiliriz.
- Var olan bir grubu silmek istediğiniz zaman ise, «groupdel grup_ismi» komutunu kullanabilir veya /etc/group dosyasından ilgili satırları elimizle silebiliriz.
- Bir kullanıcının hangi gruplara üye olduğunu görmek için şu komutu kullanırız.
 - groups kullanıcı_ismi

Kullanıcı ve Grup Yönetimi

- Komut satırında grupların birine kullanıcıyı üye yapmak için en kolay yol, /etc/group dosyasını düzenlemektir.
- Örnek olarak «kullanici» adlı kullanıcıyı «apache» grubuna eklemek için group dosyasını açıp «apache:x:48:webprogrammer,kullanici» biçiminde eklemek isteğinizi yerine getirecektir.

Kullanıcı ve Grup Yönetimi

- Bir kullanıcının birden fazla grubun üyesi olabileceğini daha önce belirtmiştik.
- Herhangi bir sebepten dolayı kullanıcı kimliliğinizi değiştirmeden etkin grubu değiştirmek isteyebilirsiniz.
 - Tipik olarak bu erişim yetkileriyle alakalı bir durumdan kaynaklanıyor olabilir.
- Bu durumda «newgrp grup_ismi» komutu işinizi görecektir.
 - Eğer bir gruba geçmek için parola sorulması gerekiyorsa, newgrp komutu bu parolayı da soracaktır.

Disk Bölümü ve Dosya Sistemi Yönetimi

- Disk bölümlerini ve dosya sistemlerini yönetmek, sık sık yapılmasa da sistem yönetiminin önemli bir parçasıdır.
- Bu bağlamda, kabuk ortamında kullanılan önemli komutlar şunlardır:
 - fdisk
 - mkfs

Disk Bölümü ve Dosya Sistemi Yönetimi

- «fdisk» disk bölümleri üzerinde birçok işlem yapmamızı sağlayan bir komuttur. Bölümler hakkında detaylı bilgiler alabilir, bölümleri bölebilir, birleştirebilir, ayrıca disklerin sektör ve silindir değerlerini değiştirebiliriz.
 - Bu komut sistem yöneticisi yetkileri ile çalıştırılan bir komuttur.
- «fdisk» komutunu kullanmak sistemde kalıcı değişiklikler yapacağı için dikkatli kullanılmalıdır.
- Disklerin ve disk bölümlerinin listelenmesi için «fdisk -l» komutu kullanılır.

Disk Bölümü ve Dosya Sistemi Yönetimi

- Belirli bir disk bölümünün biçimlendirilmesi için ise «mkfs» komutu kullanılır.
 - `mkfs [disk bölümü] -t [dosya sistemi]`
 - Örneğin `mkfs /dev/sdc1 -t ext4`
- Disk bölümleri kavramı USB bellek ünitelerini de içermektedir. Bir USB bellek ünitesini yeniden biçimlendirmenin en hızlı yolu mkfs komutunu kullanmaktır.

Disk Bölümü ve Dosya Sistemi Yönetimi

- Disk bölümleri ve dosya sistemlerinde oluşabilecek arızaların tespit edilmesi, onarılması ve (bir olasılıkla) gerçekleşen veri kayıplarının geri alınması için «fsck» komutu kullanılır.
 - fsck [disk bölümü] [seçenekler]
- Bu komutun kullanımı ext3 ve benzeri günlüklü dosya sistemleri ile birlikte önemli ölçüde azalmıştır.
 - Komut ile bağlı dosya sistemleri üzerinde işlem yapılmaz. Bu yapılacak olan işlemlerin türleri nedeni ile anlamlı bir kuraldır.

Yetkilendirme modeli

- Linux'da bir kullanıcının kimliğinin doğrulanması ve o kullanıcının yetkilendirilmesi, Linux'un bütüm güvenlik mekanizmaları için önemli bir ön adımdır.
- Linux'un temel kullanıcı doğrulama mekanizması, eklenebilir doğrulama modülleri (İng. Pluggable Authentication Modules – PAM) adı verilen bir bileşen mimarisini kullanır.
- Bu mimari sayesinde Linux içerisinde çalışan herhangi bir süreç, dinamik olarak kullanıcının kimliğini doğrulaması ve yetkilerinin kontrol edilmesi işlemini PAM alt yapısına yaptırabilir.
- Ayrıca PAM ayar dosyaları kullanılarak istenen uygulamaların bu tür denetimler talep etmesi dahi engellenebilir.

Yetkilendirme modeli

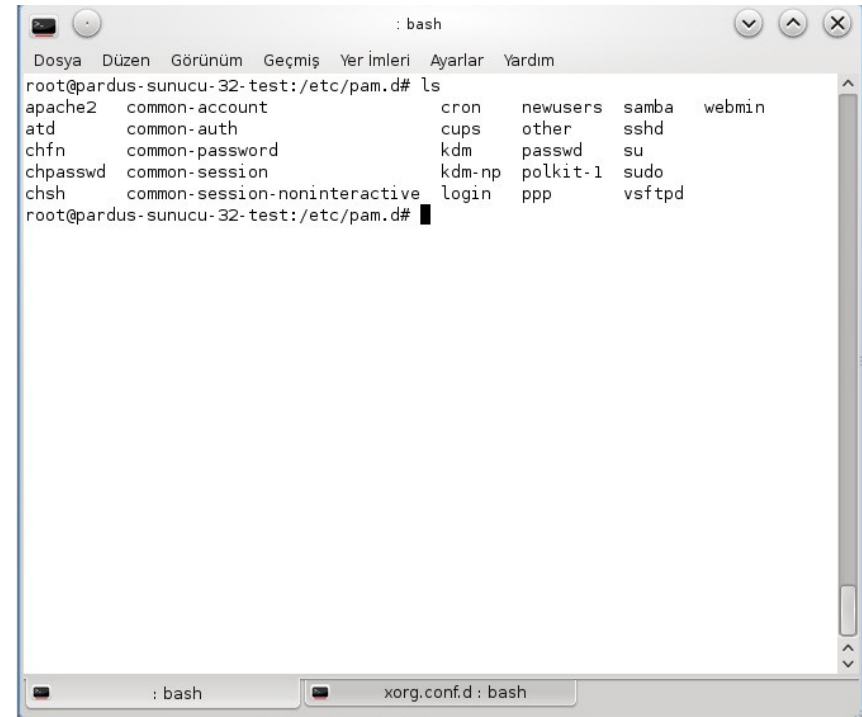
- PAM kullanan basit bir uygulama, aşağıdaki adımları takip etmektedir.
 - Uygulama içerisinde PAM kitaplıklarını (libpam.so) yükleyip ön ayarlarını yapar.
 - PAM modülleri için yetkilendirme talep eder ve olası hatayı (PAM'ın bu talebi reddetmesi) yönetir.
 - Kullanıcı kimliği ve hesap detaylarını kontrol eder.
 - Bir PAM oturumu açar.
 - Kullanıcı için çevreyi yetkilere uygun biçimde düzenler.
 - Kullanıcının işi bittiği zaman çevreyi eski haline dönüştürür.
 - PAM oturumunu kapatır.
 - PAM kitaplığı ile bağını keser.
 - Uygulama kendiliğinden kapanır.

Yetkilendirme modeli

- PAM için dört tür kullanıcı doğrulama yapılabilir
 - Kullanıcı hesabı modülleri, «hesap doğrulama» (bu hesap gerçekten var mı, hesap için bir zaman aşımı gerçekleşmiş mi, bu hesabın temsil ettiği kullanıcı, doğrulama talebini yapan uygulamayı çalıştırmaya yetkili mi, vb sorular) işlemi yapmaya yarar.
 - Doğrulama modülleri, kullanıcının kimliğini doğrulaması için gereken işlemleri (kullanıcı adı ve parola sormak) yapmaya yarar.
 - Parola modülleri, parolaları güncellemeye yarar.
 - Oturum modülleri, kullanıcı hesapları için bir oturum oluşturup o oturumu yönetmeye yarar.
- Burada «modüller» diyerek çoğul ifade edilmesi bilinçli ve önemli bir detaydır. Bu dört başlık aslında birer kategoridir.

Yetkilendirme modeli

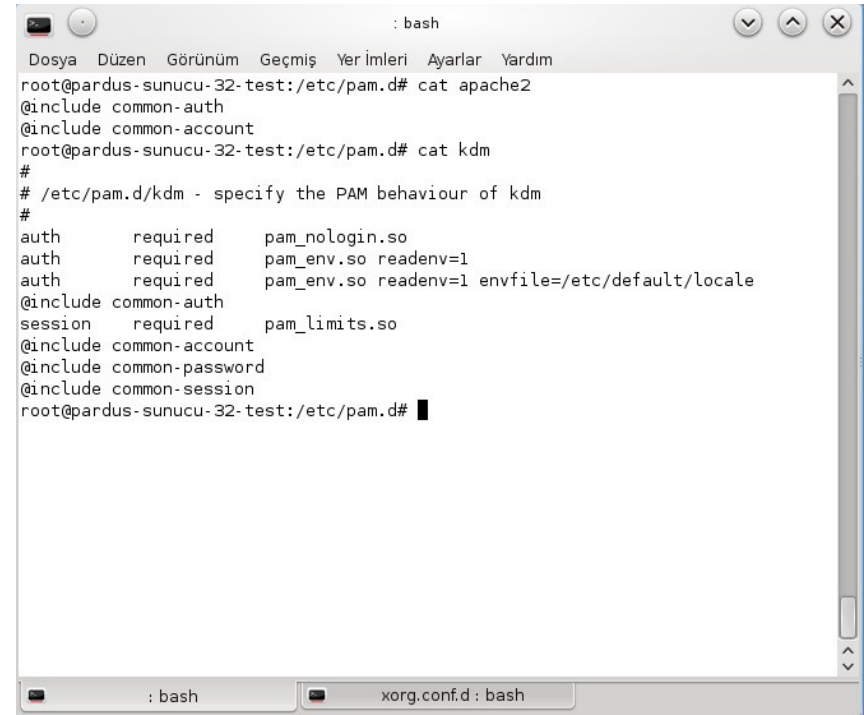
- PAM ayar dosyaları /etc/pam.d dizini altında yer almaktadır.
 - PAM kullanarak kullanıcıların kimliklerini doğrulayan servis sayısı arttıkça bu dizindeki ayar dosyası sayısı da artacaktır.
 - Her uygulamanın PAM'dan beklentisi farklı olduğu için ayar dosyaları da farklı düzeyde karmaşıktır.



```
root@pardus-sunucu-32-test:/etc/pam.d# ls
apache2  common-account      cron    newusers  samba  webmin
atd      common-auth         cups    other     sshd
chfn     common-password     kdm     passwd    su
chpasswd common-session       kdm-np  polkit-1  sudo
chsh     common-session-noninteractive login    ppp      vsftpd
```


Yetkilendirme modeli

- Pardus 2011.3 RC1 sunucu sürümünde Webmin adlı uygulama kurulu geldiği için Apache web sunucusu sistem yöneticisi kimliğini doğrulamak zorundadır.
 - Bu doğrulama PAM üzerinden yapıldığı için /etc/pam.d dizininde apache2 adında bir ayar dosyası bulunmaktadır.
- PAM'ı en çok kullanan araçlardan ikisi KDM ve GDM masaüstü yöneticileridir. Çünkü sisteme giriş yaparken kullanıcı adı ve parolasını alan ve PAM üzerinden doğrulatan araçlar bunlardır.



```
: bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
root@pardus-sunucu-32-test:/etc/pam.d# cat apache2
@include common-auth
@include common-account
root@pardus-sunucu-32-test:/etc/pam.d# cat kdm
#
# /etc/pam.d/kdm - specify the PAM behaviour of kdm
#
auth        required      pam_nologin.so
auth        required      pam_env.so readenv=1
auth        required      pam_env.so readenv=1 envfile=/etc/default/locale
@include common-auth
session     required      pam_limits.so
@include common-account
@include common-password
@include common-session
root@pardus-sunucu-32-test:/etc/pam.d#
```

Yetkilendirme modeli

- Linux PAM ayarları, PAM kullanan bir uygulama kendi PAM oturumunu açtığı sırada okunur.
 - Dolayısı ile ayar dosyaları gün içerisinde sık sık yeniden okunacaktır.
 - Eğer /etc/pam.d dizini mevcut ise PAM /etc/pam.conf dosyasını aramaz ve /etc/pam.d dizini altındaki dosyaları okur.
- Linux PAM, bir arıza durumunda erişim haklarını en aza indirecek yöne doğru ayarlıdır.
 - Dolayısı ile PAM ayarlarındaki bir hata, PAM'ın bir çok şeye erişimi engellemesi ile sonuçlanabilir.
 - Özellikle PAM ayar dosyalarının yanlışlıkla silinmesi durumunda, PAM'ın davranışı tüm erişim haklarını ortadan kaldırarak sistemi kilitleyecek şekilde olacaktır.

Yetkilendirme modeli

- Örnek olarak aşağıdaki PAM girdisi incelenebilir.

```
squid auth required pam_mysql.so user=passwd_query  
passwd=mada \
```

```
db=eminence [query=select user_name from  
internet_service where \
```

```
user_name='%u' and password=PASSWORD('%p') and \  
service='web_proxy']
```

Sistem yönetim komutları

- Linux komutları genelde seçenekler ve argümanları oluşturan bir veya daha fazla sözcükten oluşur. Komutların genel söz dizimi şu şekildedir:
 - Komut [seçenekler] [argümanlar]
- Hiçbir argüman yada seçenek almadan çalışan komutlar da mevcuttur.
- Bir çok Linux komutu genelde yapılan işin İngilizce kısaltmasıdır.
- Linux komutlarının adları hemen her zaman tamamen küçük harflerden oluşur. Ancak seçeneklerinde veya argümanlarında büyük harfler bulunabilir
- Bilgisayar kitaplarında baskı hataları ile komut adları büyük harf ile yazılabilir, dikkat edin.

Sistem yönetim komutları

Yardım Almak - man

- Man komutu, arkasından adı argüman olarak girilen komutla ilgili kullanıcı kılavuzuna erişmemizi sağlar.
- man komut_ismi şeklinde kullanılır.
- Mesela, man cd komutunu yazdığımız zaman cd komutunun kullanım kılavuzuna erişmiş oluruz.

Yardım Almak - apropos

- Apropos komutu ise verilen anahtar kelimeler için, sistem komutlarının kısa açıklamasını içeren veritabanında arama yapar ve sonuçları gösterir.
- apropos anahtar_sözcük şeklinde kullanılır.
- Örneğin, apropos pstree komutu ile tek satırlık bir kısa açıklama gelir.

Sistem yönetim komutları

Dosya İşlemleri – ls

- ls komutu dizinleri listelemek için kullanılan çok yaygın bir komuttur.
- Herhangi bir parametre vermezseniz ls dizin_ismi şeklinde çalışır.
- Komutun detaylı bilgiler vermesi için -l tipik bir parametredir.

Dosya İşlemleri - pwd

- pwd komutu an bulunulan yolu verir.
- Bu komutun çıktısı aynı zamanda Linux kabuğunda nokta (.) ile ifade edilen, o anda bulunduğunuz konumu da verir.

Sistem yönetim komutları

Dosya İşlemleri – cd

- İçinde bulunduğunuz dizini değiştirmek için cd komutunu kullanabilirsiniz.
- Sadece cd komutu sizin ev dizininize gider.
- Eğer parametre olarak başka bir konum verirsek, o konuma gider.
- Doğal olarak bu konumun bir dizin olması gerekir.
- Örnek: cd /home/kullanici/Desktop/

Dosya İşlemleri - head

- Zaman zaman bir dosyanın ilk birkaç satırına bakıvermek isteriz.
 - Bu durumda head komutunu kullanmak yeterli olacaktır.
- Örnek olarak liste.txt dosyasının ilk 5 satırı için
 - head -5 liste.txt
- head komutuna bir argüman vermezseniz, ilk 10 satırı yazdırır.
 - Parametre olarak -c byte_miktarı yazarak dosyanın istenilen uzunluktaki bölümü listelenir. Yalnız bayt sayısı ile gösterilen satır sayısı arasında doğrudan bir ilişki olmaz.

Sistem yönetim komutları

Dosya işlemleri - tail

- Bir dosyanın en son birkaç satırına bakmak da mümkündür. Örneğin, sistem kayıt günlüklerinin son satırları en son olan olaylara ait olan satırlardır.
 - Bunun için tail komutunu kullanırız.
- Bu komutun kullanımı head ile aynıdır. Örnek olarak:
 - tail -5 liste.txt

Dosya işlemleri - cat

- Birkaç dosyayı ardışık olarak yazarak birleştirmek veya ekrana yazmak istersek cat komutunu kullanırız.
- Bu arada okunan dosyalarda bir değişiklik olmaz.
- Aslında ekran da bir dosya olarak görülebildiği için cat'in sonuçları ekrana yazdırması anlaşılmasa da değildir.

Sistem yönetim komutları

Dosya işlemleri – cut

- Bir dosyanın belli alanlarını alıp yazdırmak için cut komutunu kullanırız.
- Bu özellikle boşluk veya sekme (tab) karakterleri ile ayrılmış sütunlar şeklindeki dosyalarda anlamlıdır.
 - cut -c1-6 veriler.txt
- ile dosyadaki her satırın 1 numaralı sütunundan 6 numaralı sütuna kadar olan parçayı alır ve yazdırır.

Dosya işlemleri - cut

- cut komutunu, kullanırken şu parametreleri kullanabilirsiniz:
 - -c ile sadece karakterler seçilir
 - -d ile ayırıştırıcı kullanılabilir
 - -f ile istenilen alanlar seçilebilir

Sistem yönetim komutları

Dosya işlemleri - paste

- Benzer bir komut olan paste komutu ise dosyaların satırlarını birleştirir.
- `paste dosya1 dosya2 > dosya3` komutu dosya1 ve dosya2'nin her satırını birleştirir ve file3 ün içerisine kaydeder.
- Bu arada dikkat edilmesi gereken şey şudur. İki dosyada eşit sayıda satır olmazsa, dosya3'ün son satırları sadece tek dosyadan gelen verileri içerir.

Dosya işlemleri - sort

- `sort` :Dosyadaki veriyi kullanıcının ihtiyaçlarına göre sıralar
- `sort günlük.txt` dosyanın içeriğın a'dan z'ye sıralar
- `sort` komutunu,
 - `-n` ile kullandığımız zaman sayıya göre sıralan yapar
 - `-d` ile boşluklar ve alfanümerik değerlere göre sıralanır
 - `-r` ile tam ters sıralama elde etmiş oluruz

Sistem yönetim komutları

Dosya işlemleri – cp

- Dosya ve dizinleri, farklı konumlara kopyalamak için cp komutu kullanılır.
- Örnek olarak
 - cp dosya1 /home/kullanıcı/ komutu dosya1 dosyasını kullanıcı dizinine kopyalar.
 - cp dizin1/* dizin1_yedek komutu, dizin1 içinde bulunan tüm dosyaları dizin1_yedek dizini içine kopyalar.

Dosya işlemleri – cp

- cp komutunu,
 - -f ile kullanarak var olan hedef dosya açılmadığı takdirde siler ve yeniden kopyalamayı dener
 - -i ile silinme onayı istenmesi sağlanır
 - -R veya -r ile dizinlerin içerikleri ile altdizinlerini kopyalamayı sağlar

Sistem yönetim komutları

Dosya işlemleri – mv

- Benzeri bir komut da dosyaları taşımaya yarayan mv komutudur.
- Linux komut satırında bir dosyanın adını değiştirmek, onu aynı dizinde farklı adlı bir dosyaya taşımak demektir. Dolayısı ile dosyayı bir başka dizine farklı adla taşımak da tek komut ile yapılabilir.

Dosya işlemleri – mv

- Örnek olarak:
 - mv dosya dizin1/ komutu ile dosyayı dizin1 adlı dizine taşırız.
 - mv dosya yenisosya komutu ile dosyanın adını değiştirmiş oluruz.
 - mv dosya dizin1/yenisosya komutu ile dosyanın hem konumunu hem adını değiştirmiş oluruz.

Sistem yönetim komutları

- Linux kabuğu için klavyeden verilen girdi (standart girdi) ve ekrana verilen çıktı da birer dosyadır. Bu nedenle bu girdi/çıktıyı işlemek için komutlar da bulunur.
 - Bu amaçla en basit komut tee komutudur.
- Komut, standart girdiden okur ve standart çıktıya yada dosyaya yazar.
 - Örnek olarak `ls | tee dosya_adları.txt` komutunda, `ls -l` komutu bulunulan dizinin içerisini listeler ve ekrana yazdırır. Boru (`|`) ile yakalanan bu çıktıyı tee komutuna verdiğimizde, tee bu çıktıyı dosyaya kaydeder.
- tee komutu parametresi olarak,
 - `-a` ile verilen dosyalara ekleme yapıp, üzerine yazmaz

Sistem yönetim komutları

- Dosyalar arasında bağlantı (link) kurmak için `ln` komutu kullanılır.
 - Bu bağlantılar Windows'daki kısayollar gibidir ancak Linux'da daha kapsamlı kullanımlar mümkündür.
- Bağlantıların kendileri ayrı birer dosyadır ve özgün dosyaya farklı bir konumdan erişmenizi sağlar.
 - Basit bir örnek olarak, `ln dosya1 dosya2 dizin1` komutu ile, `dosya1` ve `dosya2` dosyaları `dizin1` adlı dizine bağlanır.
- `ln` komutunu,
 - `-f` ile var olan hedef dosyalarını silmeye yarar
 - `-b` ile var olan hedef dosyalarının hepsini yedekler

Sistem yönetim komutları

- İlgili çekici bir komut olan lsof kullanımında olan her bir dosyanın isminin listesini verir.
 - Örneğin, lsof -i TCP:80 komutu 80 numaralı kapı üzerindeki tüm TCP dosyalarını gösterir.
- Bu çıktının biçimlendirilmesi zayıftır. Zaten lsof çıktısını genelde başka kabuk komutlarına aktarmak için kullanırız.
- lsof komutunda bilinmek istenen açık dosyalar, süreçler tarafından açılan disk dosyaları, ağ soketleri ve aygıtlar olabilir

Sistem yönetim komutları

- Belli bir dosyayı açmış bulunan uygulamaların (süreçlerin) listesi için fuser komutunu kullanırız.
 - Örneğin, fuser bilgi.txt komutu bilgi.txt dosyasını açan tüm süreçleri listeler
- fuser komutunu,
 - -k ile dosya erişimi yapan süreçler öldürülür
 - -l ile tüm bilinen sinyal isimlerini listeler
- fuser komutu sadece PIDleri standart çıktıya yazar, aksi takdirde herşey stderr(standart hata)' a gönderilir

Sistem yönetim komutları

- Dosyaları ve dizinleri silmek için sırası ile `rm` ve `rmdir` komutlarını kullanırız.
- `rm` komutunu,
 - `-f` ile silmeye zorlanır
 - `-i` ile silinip silinmeyeceği hakkında onay istenir
 - `-r` veya `-R` ile dizin ve alt dizinlerde bulunan tüm dosya ve dizinler silinir
- `rmdir` komutunu,
 - `-p` ile bir üst dizin ile birlikte siler

Sistem yönetim komutları

- Dizin hiyerarşisinde bir çok disk bölümünün bulunabileceğini söylemiştik.
- Bu disk bölümlerinin hangisinin nerede bağlandığı ve ne büyüklükte olduğu, bu alanın ne kadarının kullanımda olduğunu takip etmek önemlidir.
- Bazı dosya sistemlerinde boş alan kalmaması sorunlara yol açacaktır.
- Bu amaçla df komutunu kullanırız.
- df komutu parametreleri:
 - -h ile okunulabilir olarak dosya boyutları kb,mb cinslerinden görüntülenir
 - -p ile POSIX çıkış formatı kullanılır
 - -T ile dosya sistemi tipi listelenir

Sistem yönetim komutları

- Disk bölümleri ve dosya sistemlerinden bahsederken sürekli olarak bir disk bölümünün (örneğin USB üzerinden bağlanan bir harici diskin) bağlanmasından, bağlı olmasından bahsettik.
- Bu işlem sistem yöneticisi yetkileri ile yapılan önemli bir işlemdir.
- Dizin hiyerarşimize, bir dosya sistemi bağlamak için mount komutu kullanılır.
 - Genellikle mount aygıt dizin şeklinde kullanılmaktadır.
 - Mesela, `mount /dev/sdb /home/bora/Desktop/tmp_usb` komutu `/dev/sdb` ile simgelenen bir aygıtı, `tmp_usb` dizinine bağlar.

Sistem yönetim komutları

- Önceden bağlanmış bir dosya sistemini hiyerarşiden ayırma işlemine `çözme` adı verilir.
- Bağlantıyı `çözme` işlemi için `umount` komutu kullanılır.
- Bu komut da yine sistem yöneticisi yetkileri gerektiren bir komuttur.
 - Komutu `umount` dizin şeklinde kullanırız.
- Bağladığınız dosya sistemleri üzerinde işlemler yaparken o işlemler sürdüğü sırada bağlanan bölümü `çözmeniz` riskli bir durumdur. Veri kaybına neden olabilir. Bu nedenle bu işlem normalde izin verilmeyen bir işlemdir.
 - `Umount` komutuna `-f` (force, zorla) parametresini ekleyerek bu işlemin yapılmasını istediğinizi belirtebilirsiniz. Bu durumda dahi bazı teknik nedenler ile hata almak mümkündür. Bu özellikle yerel ağ üzerinden paylaşılan dizinlerde karşılaşılabilen bir sorundur.

Sistem yönetim komutları

- Zaman zaman bir sistemin aşırı yavaşlamasının nedenini teşhis etmek veya sistemde başlatılacak bir uygulama için yeterli kaynak olup olmadığını anlamak için bir çalışma yapmak gerekebilir.
- Bir diğer yararlı kabuk komutu olan free, sistemdeki takas ve fiziksel bellekte bulunan boş veya kullanılmayan alanlar hakkında bilgi verir.
- Genellikle free yazarak tek başına kullanılır.
- free komutunu kullanırken:
 - -b ile çıktığı bayt cinsinden gösterebiliriz
 - -k ile kilobayt cinsinden
 - -m ile megabayt cinsinden
 - -g ile gigabayt cinsinden
 - -l düşük ve yüksek bellek kullanımı hakkında detaylı bilgi verir

Sistem yönetim komutları

- Sistemdeki toplam yükü ve bu yükü oluşturan kullanıcılara ait faaliyetleri görmek için w komutunu kullanırız .
- Genellikle bir parametre olmadan sadece w yazarak tek başına kullanılır.
- w komutunu,
 - -h başlığı göstermez
 - -s ile kısa format kullanılır
 - -u ile kullanıcı ismini göstermez(süper kullanıcı olarak kullanmanız gerekmektedir)

Sistem yönetim komutları

- Bir sistemin üzerindeki yük konusunda merak edebileceğimiz başka şeyler de vardır.
 - Sistemin ne kadar zamandır çalışmakta olduğu bilgisini almak için uptime komutunu veririz.
 - Tek başına uptime yazılarak kullanmak yeterli bilgiyi verecektir.
- Sisteme giriş çıkış yapan kullanıcıları ve bunların son işlemlerini takip etmek de hata giderme için önemli bir adımdır.
 - Sisteme en son giriş yapanları görmek için last komutunu kullanırız.
 - Sadece last komutunu kullanarak detaylı bilgi elde edebilirsiniz.
 - Ayrıca, last -n 3 gibi parametre vererek, son 3 girişi de görebiliriz.
- last komutunu,
 - -a ile kullanarak son sütunda host ismini gösterebiliriz
 - -i ile uzaktaki hostun ip'sini gösterebiliriz
 - -x ile sistemin kapatılma girişlerini ve çalışma seviyeleri değişimlerini gösterebiliriz

Sistem yönetim komutları

- Kullanıcı parolalarını değiştirmek için `passwd` komutunu kullanırız.
 - Yalnızca `passwd` yazarak tek başına kullanılırsa o anki kullanıcının parolası değiştirilir.
 - Kullanıcı adı verilerek, `passwd kullanıcıadı` şeklinde girilirse o kullanıcıya ait parola güncellenir.
 - Sistem yöneticisi (root) yetkileri olan kullanıcılar başka kullanıcıların parolalarını değiştirebilir.

Sistem yönetim komutları

- O anda sistemde kimin giriş yaptığını görmek için who komutunu kullanabiliriz.
- Bu özellikle last komutu ile birlikte kullanılacak bir komuttur.
- Komut satırında who komutunu kullanırken değişik parametreler kullanmak mümkündür.
 - -b ile son sistemin başlangıç zamanını verir
 - -d ile ölü süreçler gösterilir
 - -l ile sisteme giriş süreçleri gösterilir
 - -q ile tüm login isimleri ve giriş yapan kullanıcı sayısı gösterilir
 - -u ile sisteme giriş yapan kullanıcılar listelenir.

Sistem yönetim komutları

- Diskleri bölümlendirmek için fdisk komutunu kullanırız.
 - Komut satırında sistem yöneticisi yetkilerine sahip kullanıcılar tarafından çalıştırılır.
 - Kullanımı fdisk aygıtadı şeklindedir.

Sistem yönetim komutları

- Belli bir dosyayı aradığımız zaman birden fazla güçlü aracımız vardır.
 - Find komutu yaklaşık isimlerini bildiğinizi farz ederek, dosya yada dizin araması yapar.
 - Örneğin, `find -name 'sayfam.html'` komutunu girdiğimiz zaman, ismi `sayfam.html` olan dosyaları listeler.
 - `slocate` ise dosya ve dizin bulmanın en kolay ve en hızlı yöntemidir.
 - Örneğin, `slocate dosya1` komutunu verdiğimiz zaman, ismi `dosya1` olan tüm yolları listeler.
 - `slocate` komutu sisteminizde yüklü olmayabilir.

Sistem yönetim komutları

- Xargs komutu, standart girdiden komut satırları inşa edip çalıştırmak için kullanılır.
- Genelde diğer komutlar ile birlikte kullanılır. Başka bir komutun çıktısını boru (pipe) ile xargs'a aktararak bir satır ile bir çok komutu topluca oluşturup çalıştırmış olursunuz.
- Örneğin, `find ~ -type f -mtime +1825 |xargs ls -l` komutunu girdiğimiz zaman, 5 yıldan önce ve daha fazlası yıllarda modifiye edilmiş dosyaları listeler.

Sistem yönetim komutları

- Komut satırındaki komutu sonuna & karakteri koyarak görevi arka planda çalıştırabiliriz
- Eğer ki bir görev başlatılmış ve ön planda çalışıyorsa, görevi durdurmada arka plana aşağıdaki gibi taşıyabilirsiniz;
 1. Ctrl+z tuş kombinasyonu kullanarak ön planda çalışan görevi askıya alabilirsiniz.
 2. Görev kontrol komutunu(,yani bg komutu) giriniz.
 3. Enter tuşuna basınız.
- Böylelikle görevi arka plana taşımış oldunuz.

Sistem yönetim komutları

- Gzip komutu ile, ismi verilen dosyalar Lempel-Ziv(LZ77) algoritması ile sıkıştırılır.
 - Genellikle gzip dosya_adi şeklinde kullanılır.
- Gzip ile sıkıştırılan dosyaları açmak için ise gunzip komutu kullanılır.
 - gunzip dosya_adi.gz şeklinde dosya açılır.
- Bzip2 komutu, Burrows-Wheeler blok sıralama ve Huffman algoritması ile sıkıştırır.
 - Örnek olarak bzip2 dosya_adi şeklinde kullanılır.
 - Çoğunlukla gzip'den daha iyi sıkıştırma kabiliyetine sahiptir.
- Bunzip2 komutu ile bzip2 ile sıkıştırılan dosyayı açarız.
 - Genelde bunzip2 dosya_adi.bz2 şeklinde kullanılır.

Sistem yönetim komutları

- Tek tek dosyaları sıkıştırmak yerine bir çok dosyayı topluca arşivlemek ve özellikle başka bir yere taşımak istediğimiz zaman birden çok dosyadan tek ve büyük bir dosya oluşturmak gerekir.
- Tar (teyp arşivleyicisi – tape archiever) arşivleme aracının GNU sürümü bu işe yarar.
 - Genel kullanımı tar [-xtcvzf] arşivadı.tar dosya(dizin)adı şeklindedir.
 - Mesela, tar -xvf arşiv.tar dizin şeklinde girdiğimiz zaman, arşivi bulunan dizine açar.
 - Tar arşivlerini sıkıştırdığınız zaman .tar.gz veya .tgz dosya adı uzantıları oluşur.

Sistem yönetim komutları

- Unzip komutu, zip arşivlerini açmak için kullanılır.
 - Genel olarak unzip arşiv_adi.zip -d dizin_adi şeklinde kullanılır. Böylelikle zip arşivlerini verilen dizine açmış oluruz.
- Unarj komutu ise arj arşivlerini açmak için kullanılır.
 - unarj e arşiv_adi.arj şeklinde kullanılır.

Sistem yönetim komutları

- Kodlama işlemi kendisi bir sıkıştırma olmasa dahi, genelde kodlanmış dosyalar ayrıca sıkıştırıldığı için sıkıştırma ile birlikte bahsedilebilir.
 - Unutmayın, kodlama şifreleme değildir. Kodlanmış verileri geri kodlamak için ek bilgiye ihtiyaç yoktur ve herkes geri kodlayabilir.
- Özellikle eposta transferi sırasında sık sık epostalar kodlanarak yollanır. Ancak bu epostaların her yerde kolayca açılıp okunmasını zorlaştırır.
 - Uuencode komutu, kodlanmış metinleri (genelde elektronik postalarda ekli dosyaları) açmak için kullanılır
 - Genelde uudecode -o hedef kaynak şeklinde kullanılır.

Sistem yönetim komutları

- Bilgisayarı kapatmak için halt komutu kullanılır.
- Tek başına sadece halt yazarak kullanmak yeterlidir.

Sistem yönetim komutları

- screen:ANSI/VT100 terminal emülasyonu sağlayıp 1 gerçek terminalden birden fazla tüm ekran sahte terminallerin çalışmasını mümkün kılar ve aynı zamanda pencereler arasında kopyalama,yapıştırma,ekran girdi-çıkıtlarını kaydedip işlemenize yardımcı olur.
- Sadece screen yazmak ekranı başlatmak için yeterlidir.
- screen içerisinde birçok pencere oluşturabilir ve aralarında geçiş yapabiliriz. Eğer ki siz başka birşey yapıyorsanız ,ve çalışan süreci bırakmak(mysql'e bağlantı mesela) istiyorsanız bu komut kullanışlı bir hal alır.

Sistem yönetim komutları

- Sıklıkla sistem yöneticisi olarak bir diğer kullanıcının sorunlarını çözeriz. Ancak bu durumlarda dahi bazen diğer kullanıcıların hesaplarına girip, bazı işleri onların kullanıcı kimlikleri ile yapmak yararlıdır.
- su komutunu kullanarak başka bir kullanıcının yerine geçmek için kullanılır. Örneğin su kullanıcı komutunu veren sistem yöneticisi o kullanıcının yerine geçer.
- Aynı komut sıradan bir kullanıcının root kullanıcısı olması için kullanılır
- Sadece su yazmak ve parola girmek, süper kullanıcı olmak için yeterli olacaktır.

Sistem yönetim komutları

- Su komutunun önemli bir parametresi – karakteridir.
- su yerine su – kullandığınız zaman, yerine geçtiğiniz kullanıcının çevre değişkenlerini miras alırsınız.
- Bunun etkisi önemlidir, özellikle kullanıcı adı belirtmeden su – diyerek root kullanıcısının yerine geçtiğinizde bunu hissedersiniz. Artık kendi özgü kullanıcınız yerine root kullanıcısının çevre değişkenlerine geçmişsinizdir.
- İçinde bulunduğunuz dizin de değişir.

Sistem yönetim komutları

- sudo :Bir komutu başka bir kullanıcı(genelde süper kullanıcı) olarak çalıştırmanızı sağla.ç
 - Mesela, sudo ls /usr/local/dizin şeklinde girdiğimiz zaman, sadece sistem yöneticisi olan kullanıcıların erişimine izin veren dizine erişmemizi sağlar.
- Sudo komutu, normal kullanıcının kendi parolasını yazmasını bekler.
 - Bu nedenle sudo herkesin kullanımına açık değildir. Sudo yapma izni olan kullanıcılar, ve kimin hangi komutlarda sudo yapabileceği ayarlanabilir.
 - Tipik bir sudo kullanımı sudo su – komutu ile su komutunu çağırmaaktır. Sudo su – ile su – arasındaki fark, sudo kullanan kişinin kendi parolasını, doğrudan su – yazan kişinin ise sistem yöneticisi (root) parolasını bilmesi gerekmesidir.

Sistem yönetim komutları

- Sıklıkla kullanılan su komutu için grafiksel bir ön uç (giriş noktası) gerektiği ortadadır.
- Bu amaçla kdesu (KDE) ve gtxsu (Gnome) komutlarını kullanırız.

Sistem yönetim komutları

Bash Programlama

- Bash
 - Komut satırını düzenlemeye izin vermesi
 - Takma isimler ve fonksiyonları desteklemesi
 - Geçmiş komutları çağırma ve düzenlemeyi sağlaması
 - Tamamı yazılmamış sözcükleri tamamlayabilmesi
 - Aritmetik işlemleri yapabilmesi
 - Arkaplanda görevlerin çalıştırılmasını desteklemesi sebebiyle Linux'ta hayatımızı oldukça kolaylaştırmaktadır.
- Bütün bunlara ek olarak Bash bir programlama dili ve yorumlayıcısı olarak da görülmelidir.

Bash Programlama

- Bir Bash betiği yazmak zor bir şey değildir.
 - Herhangi bir editör ile çalıştırılması istenen komutlar alt alta bir dosyanın içine yazılır.
- Betiği çalıştırmak daha da kolaydır, ancak Bash'in bazı özelliklerinden etkilenir.
 - Komut satırına çalıştırılmak istenen betiğin adı yazılır.
 - Betiğin bulunduğu dizin PATH değişkeni içinde yoksa `./` (içinde bulunulan dizin) sözdizimi kullanılabilir.
 - Bir betiğin çalıştırılabilmesi için dosya izinlerinden çalıştırma izninin bulunması gerekir.
 - Betiğin hangi kabuk kullanılarak çalıştırılacağı belirtilmemişse betiğin ilk satırına bakılır. İlk satır `#!` karakterleri ile başlıyorsa betik bu karakterlerden sonra yolu belirtilen kabuk kullanılarak çalıştırılır.

Bash Programlama

```
#!/bin/bash
```

Bash Programlama

- Bash'in bir betiği çalıştırması şu şekilde olur:
 - Kabuk kendisinden bir eş yaratarak verilen betiği çalıştırır ve betiğin çalışması bittikten sonra yarattığı eşi yok eder.
 - Betiğin çalıştığı ortam betiğin çalışması bittikten sonra yok edildiği için kullanılan bütün değişkenler de yok olur.
 - Eğer bir eş yaratmadan içinde bulunulan kabukta çalıştırılması isteniyorsa source komutu kullanılmalıdır.

Bash Programlama

- Hata ayıklama modunda betikten okunan bütün komutlar çalıştırılmadan önce son halleri ile ekrana yazdırılır.
 - -x seçeneği ile çalıştırıldığında bash bütün betiği hata ayıklama modu açık şekilde çalıştırır.
 - Hata ayıklama modu betiğin tamamı yerine sadece bir kısmında da kullanılabilir.
 - Hata ayıklama modunu açmak için “set -x” komutu verilir.
 - Hata ayıklama modunu kapatmak için “set +x” komutu verilir.

```
pardus@tubitak-uekae ~ $ cat ilk_betigim.sh
#!/bin/bash
echo "Bu benim ilk yazdigim betik."
VAR="VAR degiskeni"
echo $VAR
pardus@tubitak-uekae ~ $ bash -x ilk_betigim.sh
+ echo 'Bu benim ilk yazdigim betik.'
Bu benim ilk yazdigim betik.
+ VAR='VAR degiskeni'
+ echo VAR degiskeni
VAR degiskeni
pardus@tubitak-uekae ~ $
```

Bash Programlama

- Bash programlamasında değişkenler (1) çevre değişkenleri (İng. Environmental variables) ve (2) yerel değişkenler (İng. Local variables) olarak ikiye ayrılırlar.
 - Çevre değişkenleri her kabukta tanımlıdır ve programların değişkenleri paylaşmaları için kullanılmaktadır.
 - Çevre değişkenlerinin dışındaki bütün değişkenler yerel değişkenlerdir.
- Bir yerel değişkeni çevre değişkeni haline getirmek için export komutu kullanılır.
 - export DEĞİSKEN_ADI
- printenv komutu ile bütün çevre değişkenlerini ekrana yazdırılır.

Bash Programlama

- Bash tarafından özel anlam yüklenen değişkenlerdir. Bu değişkenlere değer ataması yapmak mümkün değildir.
 - \$0 Çalışmakta olan betiğin adı
 - \$1...\$9 Betiğe verilen parametrelerden belirtilen sırada olanı
 - \$@ Betiğe verilen parametrelerin tamamı
 - \$# Betiğe verilen parametrelerin sayısı
 - \$? En son çalıştırılan komutun çıkış kodu
 - \$\$ Çalıştırılan betiğin süreç numarası

Bash Programlama

- \ karakteri kendisinden sonra gelen karakterin özel anlamını yok eder.
 - Eğer \ karakterinden hemen sonra satır sonu karakteri geliyorsa bu karakter kaçırma değil satır devamı anlamına gelir. Terminale sığmayan yazıların alt satırdan devam etmesini sağlar. \ karakteri yazılan komuttan çıkarılır ve devam eden satır önceki satır ile birleştirilir.
- İki ' karakteri arasında yazılan her karakterin özel anlamı yok olur. Bütün karakterler yazıldığı gibi korunurlar.
 - İki tek tırnak karakteri arasında başka bir tek tırnak karakteri bulunamaz.

Bash Programlama

- İki “ karakteri arasında yazılan bazı karakterler özel anlamlarını korurlar, diğer bütün karakterler yazıldığı gibi korunurlar.
 - \$ ve ` karakterleri özel anlamlarını korurlar.
 - \ karakteri arkasından sadece \$ ` “ veya \ veya satır sonu karakteri geldiği zaman özel anlamını korurlar. Bu karakterlerden biri ile kullanıldığı zaman \ karakteri komut yorumlanırken komuttan çıkarılır.
 - Bunlar dışındaki bütün karakterler yazıldığı gibi korunurlar.

Bash Programlama

- `${DEĞİŞKEN}`
 - Değişken isimlerini diğer karakterlerden ayırmak için kullanılır.
- `{DEĞİŞKEN:=DEĞER}`
 - Eğer DEĞİŞKEN değişkenine daha önceden atama yapılmışsa DEĞİŞKEN'in kendi değeri kullanılır, eğer daha önceden atama yapılmamışsa (veya değeri "NULL" ise DEĞİŞKEN'in değeri DEĞER olarak atanır.

Bash Programlama

- `$(komut)`
- ``komut``
 - Bash komut cümlesini çalıştırır ve çıktısını `$(komut)` yerine kullanır.
 - Satır sonu karakterleri komutun çıktısından ayıklanır ve komut çıktısı `$(komut)` yerine konur. Eğer satır sonu karakterlerinin ayıklanması istenmiyorsa atama çift tırnak içinde yapılmalıdır.
- `$((ARİTMETİK İŞLEM))`
- `#[ARİTMETİK İŞLEM]`
 - Parantez içindeki aritmetik işlemi hesaplayarak sonucu `$((ARİTMETİK İŞLEM))` yerine koyar.
- Aritmetik operatörler C programlama dili ile aynı ve aşağıdaki gibidir.
 - `DEĞİŞKEN++`, `DEĞİŞKEN--`
 - `++DEĞİŞKEN`, `--DEĞİŞKEN`
 - `**`, `*`, `/`, `%`, `+`, `-`
 - `=`, `*=`, `/=`, `%=`, `+=`, `-=`

Bash Programlama

- Bir kelimenin bir komutun içinde geçen ilk kelime olması şartıyla bir karakter dizisi yerine kullanılmasını sağlar.
 - Alias komutu ile takma isim ataması yapılır.
 - Yapılmış olan takma isim atamaları unalias komutu ile silinir.

Bash Programlama

- Bash kalıp işleme alt yapısı, metinlerin belli kalıplara çevirerek kullanılmasını sağlar.
 - ? Herhangi bir karakter
 - * Hiçbir veya herhangi bir/birkaç karakter
 - [] Belirtilen aralıktaki bir karakter

```
pardus@tubitak-uekae ~ $ ls
Desktop  dizin11  dizin2  dosya10  dosya2  dosya4  dosya6  dosya8  scripts
dizin1  dizin12  dosya1  dosya11  dosya3  dosya5  dosya7  dosya9
pardus@tubitak-uekae ~ $ ls dizin*
dizin1:
dizin11:
dizin12:
dizin2:
pardus@tubitak-uekae ~ $ ls dosya?
dosya1 dosya2 dosya3 dosya4 dosya5 dosya6 dosya7 dosya8 dosya9
pardus@tubitak-uekae ~ $ ls dosya??
dosya10 dosya11
pardus@tubitak-uekae ~ $ ls dosya?[0-2]
dosya10 dosya11
pardus@tubitak-uekae ~ $ ls dosya*[01234]
dosya1 dosya10 dosya11 dosya2 dosya3 dosya4
pardus@tubitak-uekae ~ $ ls dosya?[^0]
dosya11
pardus@tubitak-uekae ~ $ ls dizin?[^0]
dizin11:
dizin12:
pardus@tubitak-uekae ~ $
```

Bash Programlama

- Bir «düzgün deyim» (İng. Regular Expression) bir karakter dizisinin bir kalıp olarak tanımlanmasıdır.
 - Belirtilen kalıba uyan bütün karakter dizilerinin yerine kullanılmasını sağlar.
 - Kalıp tanımı yaparken tekrarlama operatörleri (metakaracter) kullanılabilir.
- İçeriği sıkıca tanımlanmış karakter sınıflarına isimlerini kullanarak ulaşmak mümkündür.
 - [:SINIF:]

Bash Programlama

- İki köşeli parantez arasında yazılan birden fazla karakterin herhangi birinin eşleşmesi düzenli ifadenin geri kalanının eşleşmesi halinde yeterlidir.
 - Köşeli parantezden sonra gelen ^ karakteri verilen karakter sınıfını tersine çevirir.
 - İki karakter arasında tire işareti (-) kullanılarak karakter aralığı da belirtilebilir. Ancak karakterlerin sıraları sistemin o anki lokalizasyon ayarlarından alındığı için her zaman istenen sonuç elde edilemeyebilir.
- İçeriği sıkıca tanımlanmış karakter sınıflarına isimlerini kullanarak ulaşmak mümkündür.
 - [:SINIF:]

Grep

- Kabuk komutlarının çıktıları genelde oldukça uzun metinler oluşturabilir.
- Bu nedenle bu metinler içinde arama yapmak gerekebilir.
- Kabukta grep komutu ile kullanıcı tarafından belirlenen kalıplar doğrultusunda kelime yada kelime gurubu aramak mümkündür.
 - grep seçenekler kalıp dosya(lar) şeklinde kullanılır
- Mesela, `grep 'Pardus' dosya1 dosya2 dosya3` komutunu girdiğimiz zaman dosya1, dosya2 ve dosya3 dosyalarında 'Pardus' yazısını arar.
 - Diğer popüler kullanımı ise boru (pipe) mekanizmasıyla, örneğin `ls -l | grep rwxrwxrwx` gibi olur.

Sed

- Stream EDitor (Sed) bir dosya veya girdiden okunan metin üzerinde istenen dönüşümleri yaparak sonucu standart çıktıya yollayan programdır.
- Sed) bir dosya veya girdiden okunan metin üzerinde istenen dönüşümleri yaparak sonucu standart çıktıya yollayan programdır.

Sed

- Belirli bir kalıba uyan tüm eşleşmeleri bulmak ve başka bir karakter katarı ile değiştirmek için search-and-replace (s) komutu kullanılır.
 - Sed 's/kalıp1/kalıp2/' dosya
 - Sed 's/kalıp1/kalıp2/g' dosya
- Aynı dosya için birden fazla sed komutu çalıştırmak için her komuttan önce -e seçeneği konulur.
 - Sed -e 'sed_komutu1' -e 'sed_komutu2' dosya

Sed

```
pardus@tubitak-uekae ~ $ cat liste
İbrahim TORAMAN
Sergen YALCIN
İsmail GULDUREN
Batuhan KARADENİZ
Koray AVCI
Tanju COLAK
Gokhan GULEC
Ahmet DURSUN
Umit KARAN
Emre ASIK
Emre BELEZOGLU
pardus@tubitak-uekae ~ $ sed -e 's/Emre/EMRE/' -e '/GUL/d' liste
İbrahim TORAMAN
Sergen YALCIN
Batuhan KARADENİZ
Koray AVCI
Tanju COLAK
Ahmet DURSUN
Umit KARAN
EMRE ASIK
EMRE BELEZOGLU
pardus@tubitak-uekae ~ $ █
```

Awk

- Verilen girdi içinde belirli bir kalıbın varlığını arayıp o kalıpla karşılaşıldığında yapılması gereken işlemlerin belirtildiği programlama dilidir.
 - awk 'Program' dosya
- Awk içerisinde kullanılan print programı aşağıdaki adımlar ile çalışır:
 - Girdi'den bir satır okur.
 - Eğer var ise düzenli ifadeler ile filtreleme yapar.
 - FS değişkeninde belirtilen karaktere göre (öntanımlı değeri boşluk) satırı parçalar. Herbir parçayı \$1,\$2,...\$N numaralı değişkenlere yerleştirir.

Awk

```
pardus@tubitak-uekae ~ $ awk '{ print $1 " " $5 }' /etc/passwd |head
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/bin/false
news:x:9:13:news:/usr/lib/news:/bin/false
operator:x:11:0:operator:/root:/bin/bash
pardus@tubitak-uekae ~ $ awk 'BEGIN { FS=":" } { print $1 " " $5 }' /etc/passwd |head
root root
bin bin
daemon daemon
adm adm
sync sync
shutdown shutdown
halt halt
mail mail
news news
operator operator
pardus@tubitak-uekae ~ $ █
```

Awk

```
pardus@tubitak-uekae ~ $ awk 'BEGIN { FS=":"; OFS=":"; ORS="\n-----\n" } { print $1,$5 }' /etc/passwd |head
root;root
-----
bin;bin
-----
daemon;daemon
-----
adm;adm
-----
sync;sync
-----
pardus@tubitak-uekae ~ $ awk 'BEGIN { FS=":"; OFS=":"; ORS="\n-----\n" } { print "Record Number " NR ":\t" $1,$5
}' /etc/passwd |head
Record Number 1:      root;root
-----
Record Number 2:      bin;bin
-----
Record Number 3:      daemon;daemon
-----
Record Number 4:      adm;adm
-----
Record Number 5:      sync;sync
-----
pardus@tubitak-uekae ~ $ █
```

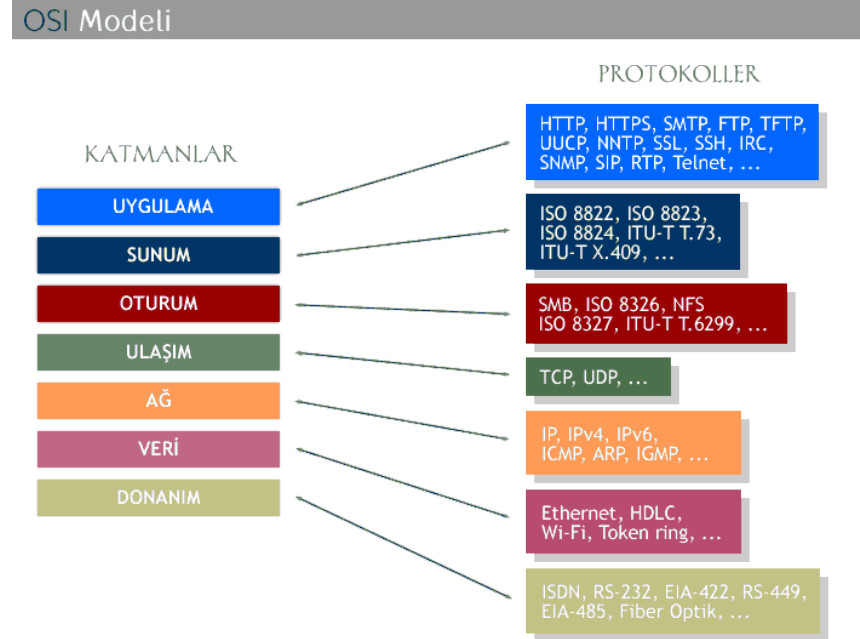
4. Bölüm

Temel Ağ Protokolleri ve Servisleri

- Bu bölümde, Pardus üzerinde kurulabilecek olan çeşitli ağ servisleri ve bu servislere ilişkin protokollerden bahsedilecektir.
- Başlıca konular şunlardır:
 - Temel Ağ Kavramları ve TCP/IP
 - SSH
 - DHCP, DNS ve NTP
 - NFS vs SMB/CIFS
 - CUPS
 - VNC ve RDP

Temel Ağ Kavramları ve TCP/IP

- TCP/IP protokollerinden bahsetmeden önce, OSI katmanlarından bahsetmek gereklidir.
- OSI katmanları ve bu katmanların görev dağılımı ISO tarafından geliştirilen ve bilgisayarlar arası iletişimin nasıl olacağını tanımlayan kurallar bütünüdür.



Temel Ağ Kavramları ve TCP/IP

- OSI 7 temel katmandan oluşmaktadır:
 - 7- Uygulama Katmanı (Application Layer): Verilerin birleştirilip çalışır duruma getirilmesi
 - 6- Sunum Katmanı (Presentation Layer): Verinin sıkıştırılma, şifreleme, formatının belirlenmesi, düzenlenmesi.
 - 5- Oturum Katmanı (Session Layer): NETBIOS
 - 4- Taşıma Katmanı (Transport Layer): TCP/IP, SPX
 - 3- Network Katmanı (Network Layer): Router
 - 2-Veri Hattı Katmanı (Data Link): Ethernet (NIC) ve Token Ring
 - 1-Fiziksel Katman (Physical Layer): Ağ içerisi kablolar ve jak'lar bu kısma girer.
- TCP/IP OSI 4. katmanda yer alır.

Temel Ağ Kavramları ve TCP/IP

Katman 7 – Uygulama Katmanı

- Kullanıcının ağ üzerinde veya yerel olarak kullandığı veya gereksinim duyduğu servislerin çalıştırılabildiği katmandır.
- Bunlar arasında tarayıcı, ftp bağlantı isteği, sunucu uygulamaları gibileri sayılabilir.
- HTTP protokolü kullanılarak çalıştırılan tarayıcı (Firefox, Opera, vs.) bu katmanda çalışır.

Katman 6 – Sunum Katmanı

- Bu katman eldeki verinin içerik olarak belirlenmesine ve üzerine yazılacak bilgilerin (header) bir üst veya alt katman tarafından bilinmesine yardımcı olur.
 - En üst katmandan bir dosyanın ağ içerisinde başka bir bilgisayara gönderildiğini düşünelim. Uygulama katmanından gelen paketlerin öncelikle uygulama katmanı tarafından eklenen bilgileri okunur. Bunlar içerik hakkında bilgi verir.

Temel Ağ Kavramları ve TCP/IP

Katman 6 – Sunum Katmanı

- Bulduğumuz katmanda verinin içeriğiyle ilgilendiği için içeriğe bakar ve ne tür bir veri olduğunu belirler.
 - Farklı işletim sistemleri farklı kodlamalar kullanır. A işletim sisteminde yazdığınız 1 rakamı B işletim sisteminde de aynı görünür.
 - Fakat işletim sistemleri farklı olunca harflere veya rakamlara verilen kodlamalar farklılık gösterir. Bu katman bu farklılığı ortadan kaldırarak her bilgisayarda da verinin aynı olmasını sağlar.

Katman 5 – Oturum Katmanı

- Bu katmanda basit olarak birden fazla bilgisayarın aynı anda sorun yaşamadan anlaşması sağlanır.
 - Aynı bilgisayardan birden fazla protokolde bir çok ağ bağlantısının kurulması için gereken oturum yönetimi bu katmanda yürütülür.
 - NETBIOS ve Soket protokolleri bu katmanda çalışır.

Temel Ağ Kavramları ve TCP/IP

Katman 4 – Taşıma Katmanı

- Bu katmanda sıkça duyduğumuz TCP, SPX, NetBEUI gibi protokoller çalışır ve genel olarak kendilerine iletilen verinin içerik olarak doğruluğunu kontrol ederler.
 - Eğer içerikte bir hata/bozulma saptanırsa düzeltmek için çalışılır.
 - Uygulama katmanı doğrultusunda kendilerine ulaşan veriler ağ içerisinde gönderilme boyutlarına bölünür.

Katman 4 – Taşıma Katmanı

- Taşıma katmanında bir çok denetim yapılır.
 - Kendisinden önceki ve sonraki katmanlar arasında verinin düzgün bir şekilde iletildiğine emin olmak için katmanlar arası eklenen başlık bilgilerinden yararlanır.
 - Çalıştığı sistemdeki veriye ait uygulamayı belirler ve karşı sistemde aynı programı bulup onunla bir bağ kurar. Bunu da sistemdeki programın verdiği başlıkları okuyarak yapar.
 - Ağ içerisinde bağlantıların kurulumu ve sona erdirilmesi yine bu katman tarafından yapılmaktadır. çakışmalarına karşı önlem alır. IP protokolünün çalışma alanıdır.

Temel Ağ Kavramları ve TCP/IP

Katman 3 – Ağ Katmanı

- Yönlendiricilere gelen veriler bu katmanda yönlendirilir.
 - Ağ içi veri akışı için yönlendiricide belirlenen tablolar vardır. Bu tablolar sabit veya değişken olabilir.
 - Ağ içi trafiğin çok yoğun olduğu yerlerde dinamik bir yapı kullanılarak “verinin en kısa zamanda” hedefine ulaşması sağlanır. Normal ağ koşullarında bu tablolar sabittir ve veri paketlerinin gideceği yollar önceden belirlenmiştir.

Katman 3 – Ağ Katmanı

- Ağ içerisindeki veri trafiği yine bu katman tarafından düzenlenir ve olası veri çakışmalarına karşı önlem alır. IP protokolünün çalışma alanıdır.

Temel Ağ Kavramları ve TCP/IP

Katman 2 – Veri Bağlantısı

- Verilerin fiziksel katmana doğru biçimde aktarılmasından sorumludur.
- Kendi içerisinde iki bölüme ayrılır: MAC ve LLC.

Katman 2 – Veri Bağlantısı

- MAC Fiziksel katmana aktaracağı veriye kendisine ve alıcıya ait MAC numarasını ekler ve karşı tarafa veriyi gönderen benim” şeklinde mesaj alır. Alıcıdan gelen bilgiler doğrultusunda veri gönderme işini düzenleyerek devam ettirir.

Temel Ağ Kavramları ve TCP/IP

Katman 2 – Veri Bağlantısı

- LLC ise alıcıya ulaşan bozuk veya hatalı verilerin tekrardan gönderilmesinden sorumludur. Üst katmanı olan Network katmanı ile iletişim halindedir ve TCP/IP protokolleri vasıtasıyla bağlantı sürekli sürdürülür.
 - Alıcı bilgisayara gönderilen verilerin gönderilme sıklığı da LLC sorumluluğundadır. Çok fazla gönderip alıcıyı cevapsız bırakmak veya veriler arası sürenin uzun tutulması gibi olumsuzlukların önüne geçer.
 - Bilinen Frame Relay, PPP gibi ağ düzenekleri vardır.

Katman 2 – Veri Bağlantısı

- Yönlendirici özelliği bulunan modemlerde yer alan “Bridge Filtering” kabiliyeti, LLC ayarları içindedir.
 - Ağ bağlantılarında seçtiğimiz PPPoE LLC, 1483 Bridged IP LLC gibi protokoller bu düzenin sağlanması için yapılmaktadır. Modeminizden göz atabilirsiniz.

Temel Ağ Kavramları ve TCP/IP

Katman 1 – Fiziki Katman

- Bu katmana kadar işlemler ile gelen veri Fiziksel katmandan kablolar yardımıyla iletilir.
- Veri içerik olarak bir ve sıfırdan oluşmaktadır, bu noktadan sonra bu bir ve sıfırlar elektrik sinyallerine dönüştürülerek kablolara aktarılır ve alıcı bilgisayar gönderilir.

Katman 1 – Fiziki Katman

- Alıcı bilgisayar da aldığı elektrik sinyallerini ters işlemle bir ve sıfırlar haline getirerek yorumlayacaktır.
 - Burada önemli olan kısım alıcı ve göndericinin kullandığı voltaj ve milisaniyelerin aynı olmasıdır. Aksi durumda gönderilen veriler anlamsız olarak algılanacak ve iletişim sağlanamayacaktır.
 - Fakat günümüzde kullandığımız ethernet alt yapısı bu farklılığı ortadan kaldırarak tek bir değer kullanıp sorun yaşanmasının önüne geçmiştir.

Temel Ağ Kavramları ve TCP/IP

Ağ Topolojileri

- Doğrusal (Lineer bus),
- Halka (ring),
- Yıldız(star)

Ağ Donanımları

- Ağ arabirim kartı(NIC),
- Veri iletim ortamı,
- Anahtar,
- Yönlendirici

Temel Ağ Kavramları ve TCP/IP

- Protokol: Fiziksel hatlar üzerindeki veri akışını düzenleyen mantık.
- TCP/IP(transmission control protocol / internet protocol)
- Ortak dil.
 - Alt protokoller: TCP, UDP, ICMP ...
 - IP: IETF RFC 791
 - TCP: IETF RFC 793

Temel Ağ Kavramları ve TCP/IP

- IP Version 4
- 32 bit IP adreslemesi
- 11111111.11111111.11111111.11111111
- 00000000.00000000.00000000.00000000
- 1 1 1 1 1 1 1 1
- 128 64 32 16 8 4 2 1
- 10010000.11111010.10100110.01000010
- 144 . 122 . 166 . 66

Temel Ağ Kavramları ve TCP/IP

- Alt ağ maskesi (subnet mask)
- IP'nin hangi kısmının "ağı" hangi kısmının bu ağın uçlarını tanımladığını gösteren değer.
- $IP \text{ (AND) MASK} = \text{NETWORK}$
- 10010000.11111010.10100110.01000010
AND
- 11111111.11111111.11111110.00000000
- (255.255.255.0)
- =
- 10010000.11111010.10100110.00000000
(144.122.166.0)

Temel Ağ Kavramları ve TCP/IP

- IPv4 sınıfları
 - A sınıfı: 255.0.0.0 (IP/8)
 - B sınıfı: 255.255.0.0 (IP/16)
 - C sınıfı: 255.255.255.0 (IP/24)
- IP'ler bitiyor, bitiyor, bitti.
 - IPv6 (IPng)
- Alt ağlar
- 4 milyar IP nasıl yönlenecek? IP'ler gruplandırılır ve her bir tekil IP grubu ile hareket eder (subnetting).
 - Bir C bloğunu 4'e bölmek:
 - $256 \text{ uç} / 4 = 64$ (her bir alt ağ)
 - $256 - 64 = 192$
 - □ 255.255.255.192 subnet mask
 - 11111111.11111111.11111111.11000000 = 26 adet 1 (/26)
 - 4 Alt ağ adresi: x.x.x.{0,64,128,192}
 - Örnek: 144.122.166.73/26 ya da 144.122.166.64/26

Temel Ağ Kavramları ve TCP/IP

- IPv6
- 128 bit IP adreslemesi
- 340,282,366,920,938,463,463,374,607,431,768,211,456 farklı IP adresi

Temel Ağ Kavramları ve TCP/IP

- IPv6
 - Adres uzayı büyüyor
 - Adres kümelemesi
 - Basitleştirilmiş IP başlığı
 - Akış ve Öncelik bölümleri ile veri ayırımı, servis kalitesinin yükselmesi
 - IPv4 adreslerle beraber çalışabilirlik
 - Hop limit ile sonsuz dolanan paketler engellenir.
- Başlangıç için 3 çeşit IPv6 adresi olacak.
 - unicast: tekil noktalar
 - cluster: Ortak bir ön ek paylaşan adresler.
 - multicast: Bu grup içindeki bir adrese gönderilen paket, grup içindeki tüm noktalara ulaştırılır.
 - Unicast adresler sağlayıcı, coğrafi, IPX, yerel kullanım ve IPv4 bazlı tiplere sahiptir. IPv4 bazlı olan, sürüm 6'ya geçişi sağlamak için konmuştur.

Temel Ağ Kavramları ve TCP/IP

- IPv4 ve statik yönlendirme(routing)
- 144.122.166.66 noktası
- 144.122.166.0/24 ağı
- 144.122.166.1 geçidi (gateway)
- Bulunduğu ağın içinde kendisi bulur (kendisi ağ geçididir)
 - Diğer ağlara ulaşabilmek için tanımlı ağ geçidinden geçmek zorundadır.
 - Ağ geçidine ulaşabilmesi için de ağ geçidine doğru bir yönlendirmesi olmalıdır.

Temel Ağ Kavramları ve TCP/IP

- IPv4 ve statik yönlendirme(routing)

network	mask	gateway	interface
144.122.166.0	255.255.255.0	144.122.166.66	eth0
0.0.0.0	0.0.0.0	144.122.166.1	eth0

Temel Ağ Kavramları ve TCP/IP

- Linux ve TCP/IP
 - NIC'ler: eth(n), wan(n) ...
 - ifconfig: arabirim yapılandırıcısı
 - route: yönlendirme yapılandırıcısı
 - netstat: ağ istatistikleri, durum gözleyicisi
 - ping: test aracı
 - traceroute: paket takip aracı
 - arp: sistemdeki arp ön belleğini yönetir.

Temel Ağ Kavramları ve TCP/IP

- Ağ arabirimi konfigürasyonu
- eth(n) arabirimleri ilgili çekirdek modülü tarafından oluşturulur.
 - Çekirdek modülünü(~sürücü) yüklemek için;
 - \$ modprobe modül
 - \$ modprobe 8139to
 - Kontrol etmek için;
 - \$ lsmod

Temel Ağ Kavramları ve TCP/IP

- ifconfig
- Çoğu UNIX türevinde ağ yapılandırma aracı
- Ethernet arabirimlerine IP tanımla
- Arabirimlerini devreye koy-kaldır
 - ifconfig <interface> [[<AF>] <address>]
 - [add <address>] [del <address>] [[-]broadcast [<address>]]
 - [netmask <address>] [hw <HW> <address>] [up|down]
- ifconfig -a

Temel Ağ Kavramları ve TCP/IP

- eth0 ağ arabirimine 255.255.255.0 ağ maskeli 192.168.0.66 IP'sini ver:

```
ifconfig eth0 192.168.0.67 netmask 255.255.255.0 up
```

- eth0 ağ arabirimine 192.168.0.99 IP'sini ekle:

```
ifconfig eth0 add 192.168.0.99 netmask 255.255.255.0 up
```

- eth0:0 ağ arabirimi yapılandırmasını kaldır:

```
$ ifconfig eth0:0 down
```

Temel Ağ Kavramları ve TCP/IP

- IP yönlendirme yapılandırma aracı «route» komutudur.
- Ön tanımı ağ geçidini tanımlamaya yarar.
- Ayrıca Belli ağ bölgelerine farklı ağ geçitleri tanımlamakta kullanılabilir.

```
route [-v] [-A family]
add [-net|-host] target
[netmask Nm] [gw Gw] [[dev] If]
```

```
route [-v] [-A family]
del [-net|-host] target
[netmask Nm] [gw Gw] [metric N] [[dev] If]
```

Temel Ağ Kavramları ve TCP/IP

- Durum bilgilerini göstermek için «netstat» komutu kullanılır.

```
netstat -r -i -M -s -c -p [--tcp|-t] [--udp|-u]
[--raw|-w] [--listening|-l]
[--all|-a] [--numeric-hosts] [--numeric-ports]
```

- Parametreler
 - -r: yönlendirme tablosunu göster
 - -i: arabirimleri göster
 - -s: TCP istatistiklerini göster
 - -a: Dinlenen ya da dinlenmeyen tüm portlar
 - -l: Dinlenen portlar
 - -p: Bağlantının sahibi olan programı göster
 - -n: Tüm değerleri numerik göster
 - -c: Her saniyede çıktıyı güncelle

Temel Ağ Kavramları ve TCP/IP

- \$ netstat -r
- Kernel IP routing table
- Destination Gateway Genmask Flags MSS
Window irtt Iface
- 192.168.106.0 * 255.255.255.0 U 40 0 0
eth0
- 127.0.0.0 * 255.0.0.0 U 40 0 0 lo
- netstat -i
- Kernel Interface table
- Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK
TX-ERR TX-DRP TX-OVR Flg
- eth0 1500 0 3413 0 0 0 2594 0 0 0
BMNRU
- lo 16436 0 8 0 0 0 8 0 0 0 LRU

Temel Ağ Kavramları ve TCP/IP

- Bağlantı Durumları - İnternet bağlantıları
 - ESTABLISHED : Soket bağlantısı kurulu
 - SYN_SENT : Bağlantı kurmaya çalışıyor
 - SYN_RECV : Bağlantı isteği aldı
 - FIN_WAIT1 : Soket kapandı, bağlantı kapanıyor
 - FIN_WAIT2 : Soket kapandı, karşının bağlantıyı kapatması bekleniyor
 - TIME_WAIT : Kapandıktan sonra ağda varolabilecekler paketler bekleniyor
 - CLOSED : Soket kullanılmıyor
 - LISTEN : Soket gelen bağlantılar için dinlemede
 - UNKNOWN : Bilinmiyor
- Bağlantı Durumları - UNIX alan soketleri
 - Proto : Soket tarafından kullanılan protokol
 - RefCnt : Referans sayısı. Bu sokete bağlı programlar
 - Flags : SO_ACCEPTON(ACC, bağlanmamış soketlerde), SO_WAITDATA(W), SO_NOSPACE(N)

Temel Ağ Kavramları ve TCP/IP

- root@vm:~# netstat -lp
- Active Internet connections (only servers)
- Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
- tcp 0 0 *:1024 *:* LISTEN 661/rpc.statd
- tcp 0 0 *:sunrpc *:* LISTEN 633/portmap
- tcp 0 0 *:ftp *:* LISTEN 858/xinetd
- tcp 0 0 *:ssh *:* LISTEN 825/sshd
- tcp 0 0 127.0.0.1:34513 127.0.0.1:33778 TIME_WAIT -
- tcp 0 0 192.168.106.128:1241 212.78.77.226:2266 ESTABLISHED -
- tcp 29 0 192.168.106.128:57184 144.122.12.200:21 CLOSE_WAIT -
- tcp 0 0 192.168.106.128:34542 213.139.234.226:80 TIME_WAIT -
- udp 0 0 *:1024 *:* 661/rpc.statd
- udp 64232 0 *:bootpc *:* 542/dhcpd
- udp 0 0 *:sunrpc *:* 633/portmap
- Active UNIX domain sockets (only servers)
- Proto RefCnt Flags Type State I-Node PID/Program name Path
- unix 2 [ACC] STREAM LISTENING 1366 917/gpm /dev/gpmctl
- unix 2 [ACC] STREAM LISTENING 1429 1003/xfs /tmp/.font-unix/fs7100

Temel Ağ Kavramları ve TCP/IP

- Sistemlerin ağ bağlantıları için test aracı ping komutudur.

```
ping [ -LRUbdfnqrvVaA] [ -c count] [ -i interval] [ -l preload]  
[ -p pattern] [ -s packetsize] [ -t ttl] [ -w deadline] [ -F  
flowlabel] [ -I interface] [ -M hint] [ -Q tos] [ -S sndbuf] [ -T  
timestamp option] [ hop ...] destination
```

- Parametreler

- -a : Duyulabilir ping
- -c : Gönderilecek paket sayısı
- -i : Her paket arasında beklenilecek süre
- -p : Yollanacak paketlerin içeriğini belirler
- -r : Yönlendirme tablosunda olmayan ancak doğrudan bağlı adresler için kullanılır

Temel Ağ Kavramları ve TCP/IP

- `[bora@pardus bora]$ ping testdomain.com`
- `PING testdomain.com (207.234.129.65) 56(84) bytes of data.`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=0 ttl=48 time=251 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=1 ttl=48 time=265 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=2 ttl=48 time=258 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=3 ttl=48 time=251 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=4 ttl=48 time=258 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=5 ttl=48 time=254 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=6 ttl=48 time=260 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=7 ttl=48 time=259 ms`
- `64 bytes from 207-234-129-65.ptr.primarydns.com (207.234.129.65): icmp_seq=8 ttl=48 time=265 ms`
- `--- testdomain.com ping statistics ---`
- `9 packets transmitted, 9 received, 0% packet loss, time 9752ms`
- `rtt min/avg/max/mdev = 251.078/258.290/265.171/4.797 ms, pipe 2`
- `[bora@pardus bora]$`

Temel Ağ Kavramları ve TCP/IP

- Hedefe giderken paketlerin geçtikleri yolları görmek için «traceroute» komutu kullanılır.

```
traceroute [ -dFIrvx ] [ -f first_ttl ] [ -g gateway ] [ -i iface ]  
[ -m max_ttl ] [ -p port ] [ -q nqueries ] [ -s src_addr ] [ -t  
tos ] [ -w waittime ] [ -z pausemsecs ] host [ packetlen ]
```

- Yaygın parametreler
 - -I : UDP yerine ICMP ECHO paketleri yolla (ping usulu)
 - -r : Yönlendirme tablosunda olmayan ancak doğrudan bağlı adresler için kullanılır
 - -n : IP çözümlemesi yapma

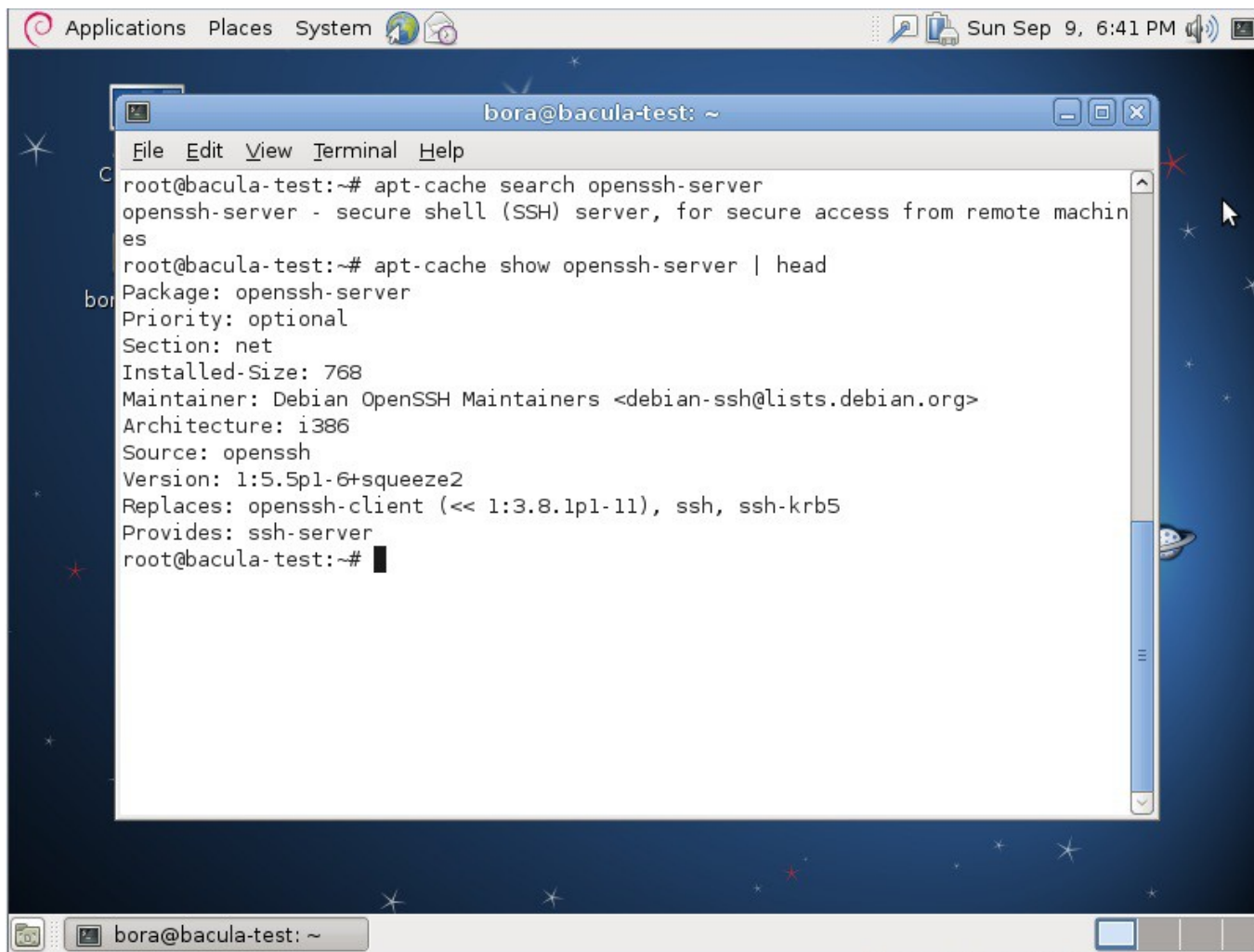
Temel Ağ Kavramları ve TCP/IP

- Bir ağdaki makinelerin IP-MAC çözümlenmesi ARP protokolü ile yürütülür.
 - 3. katmanda çalışır.
 - Ağdaki bir makineye TCP pencereleri(TCP frame) yollanacağı zaman yerel makine kendi ARP öne belleğine bakar.
 - Eğer hedef makinenin MAC adresi belleğinde ise pencereye bunu koyar ve paketi yollar.
 - Eğer ARP ön belleğinde yok ise broadcast mesaj yayınlar. Bu IP'nin sahibi olan makine bir ARP cevabı ile yanıt verir.
 - Bu broadcast mesajı ağ geçidine de gelecektir. Bu ağda olmayan bir hedefe giden bir paket varsa ve yönlendirici ağ geçidinin onunla ilgili bir yönlendirme kuralı var ise kendi MAC adresini ARP reply ile yollar.
 - `arp -a -e -s -d -n -i`

SSH

- SSH (Secure Shell), Linux sistemlere uzaktan bağlanarak kabuk erişimi elde etmenin güvenli yoludur.
- Erişim önceden belirlenmiş kriptografik anahtarlar kullanılarak şifrelenmiş bir kanaldan sağlanır.
- Linux ön tanımlı olarak OpenSSH sunucusunu kullanır.
 - Bu sunucu hemen tüm Linux kurulumlarında kurulu gelir.
 - `/etc/ssh/sshd_config` dosya yapılandırma dosyasıdır.
 - `# /bin/service openssh { start | stop | restart}`
 - `# man sshd`
- SSH sunucusu (`ssh-server`), Debian temelli sistemlerde «tasksel» alt yapısında ön tanımlı «görev» listesi içindedir.

SSH



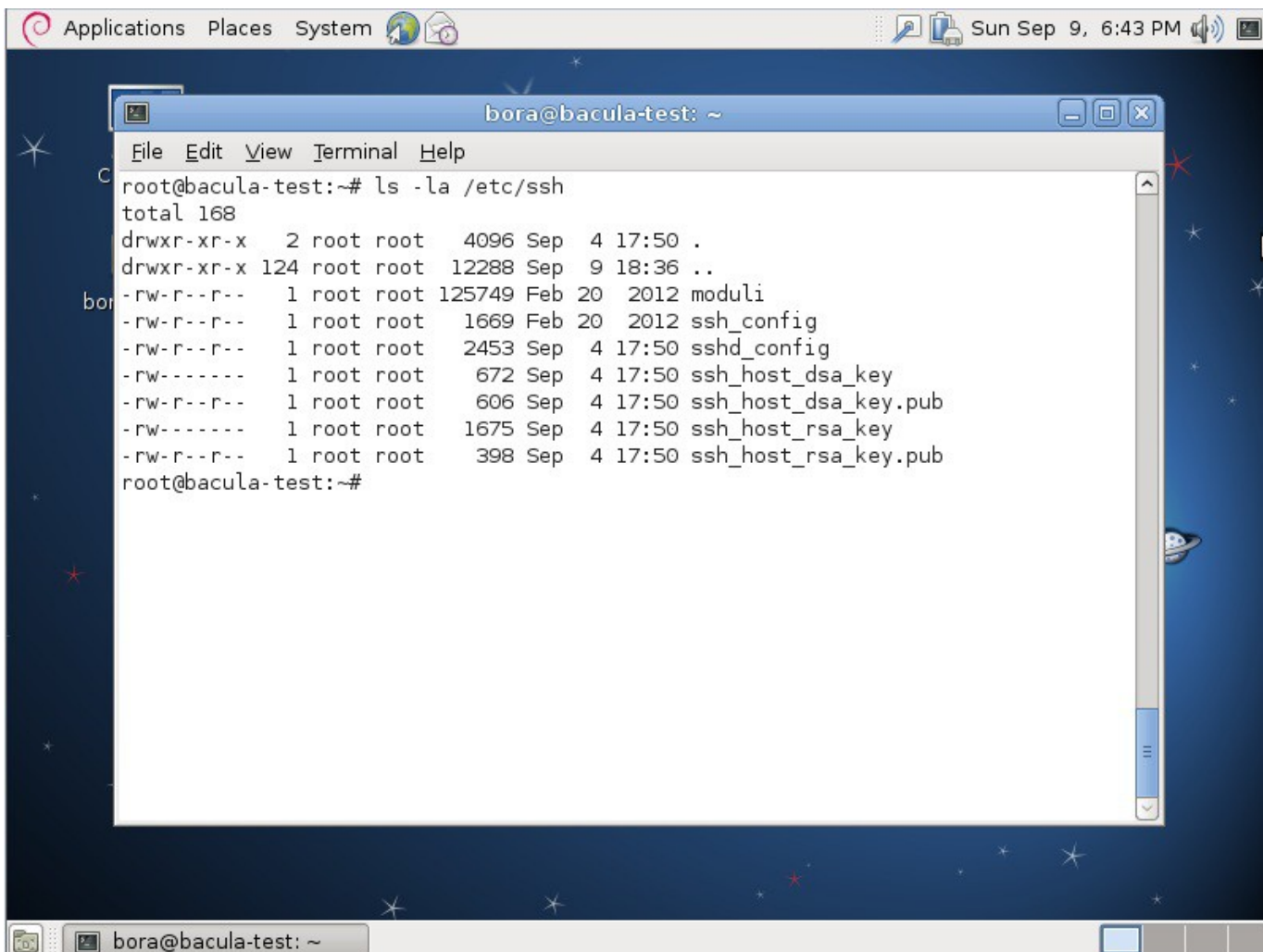
The screenshot shows a terminal window titled "bora@bacula-test: ~" with a menu bar containing "File", "Edit", "View", "Terminal", and "Help". The terminal output is as follows:

```
root@bacula-test:~# apt-cache search openssh-server
openssh-server - secure shell (SSH) server, for secure access from remote machines

root@bacula-test:~# apt-cache show openssh-server | head
Package: openssh-server
Priority: optional
Section: net
Installed-Size: 768
Maintainer: Debian OpenSSH Maintainers <debian-ssh@lists.debian.org>
Architecture: i386
Source: openssh
Version: 1:5.5p1-6+squeeze2
Replaces: openssh-client (<< 1:3.8.1p1-11), ssh, ssh-krb5
Provides: ssh-server
root@bacula-test:~#
```

The terminal window is part of a desktop environment with a dark blue background featuring stars and a rocket ship. The system tray at the bottom shows the terminal icon and the text "bora@bacula-test: ~". The top panel includes "Applications", "Places", "System", and a clock showing "Sun Sep 9, 6:41 PM".

SSH



The image shows a terminal window titled "bora@bacula-test: ~" within a desktop environment. The terminal displays the output of the command "ls -la /etc/ssh". The output lists the files and directories in the /etc/ssh directory with their permissions, owner, group, size, and modification date.

```
root@bacula-test:~# ls -la /etc/ssh
total 168
drwxr-xr-x  2 root root  4096 Sep  4 17:50 .
drwxr-xr-x 124 root root 12288 Sep  9 18:36 ..
-rw-r--r--  1 root root 125749 Feb 20  2012 moduli
-rw-r--r--  1 root root  1669 Feb 20  2012 ssh_config
-rw-r--r--  1 root root  2453 Sep  4 17:50 sshd_config
-rw-----  1 root root   672 Sep  4 17:50 ssh_host_dsa_key
-rw-r--r--  1 root root   606 Sep  4 17:50 ssh_host_dsa_key.pub
-rw-----  1 root root  1675 Sep  4 17:50 ssh_host_rsa_key
-rw-r--r--  1 root root   398 Sep  4 17:50 ssh_host_rsa_key.pub
root@bacula-test:~#
```

SSH

- Sistemdeki ssh sunucunun kullanması için yeni bir sunucu anahtarı üretmek için ssh-keygen komutu kullanılır.

```
$ ssh-keygen -q -t rsa1 -f /etc/ssh/ssh_host_key -C -N
```

```
$ ssh-keygen -q -t rsa -f /etc/ssh/ssh_host_rsa_key -C -N
```

```
$ ssh-keygen -q -t dsa -f /etc/ssh/ssh_host_dsa_key -C -N
```

```
$ chmod 600 /etc/ssh/ssh_host_key /etc/ssh/ssh_host_rsa_key  
/etc/ssh/ssh_host_dsa_key
```

```
$ chmod 644 /etc/ssh/ssh_host_key.pub
```

```
/etc/ssh/ssh_host_rsa_key.pub /etc/ssh/ssh_host_dsa_key.pub
```

SSH

- /usr/bin/ssh: SSH sunucusuna bağlanmak için kullanılır.
\$ ssh [-l login_name] hostname [command]
\$ ssh user@hostname [command]
- /usr/bin/scp: Bir sunucudan dosya ya da dizin transfer etmek için kullanılır.
\$ scp login@hostname:/yol/dosya.zip yerel_yol
\$ scp yerel_yol login@hostname:/yol/
- /usr/bin/sftp: Bir sunucuya güvenli ftp bağlantısı kurmak için kullanılır.
\$ sftp login@hostname

SSH

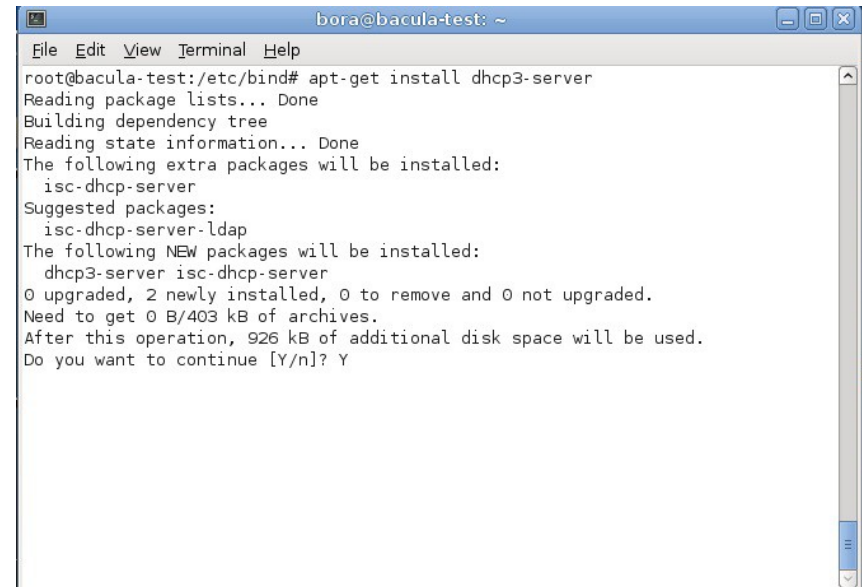
- SSH tünelleme, SSH bağlantısı kullanarak farklı protokollerin aktarımıdır.
- Yerel tünelleme: Yerel makineden SSH sunucuya doğru tüneldir.
 - `ssh user@uzakmakine -L 2000:localhost:21 sleep 99999`
 - Yerel makineden uzaktaki makinenin 2000 inci portuna bir tünel açıldı. Karşı makinedeki 21inci port bu tünelin diğer ucudur.
 - Yerelmakine : SSH : uzakmakine:2000 : TELNET : uzakmakine:21
- Uzak tünelleme: Uzak makineden yerel makineye doğru açılan tüneldir.
 - `$ ssh user@uzakmakine -R 2000:localhost:23 sleep 99999`
 - Artık uzak makine üzerinden yerel makineye bir tünel vardır.
 - `uzakmakine : TELNET : uzakmakine:2000 : SSH : yerelmakine:telnet`
 - `user@uzakmakine $ telnet localhost 2000`
 - Connected to localhost.
 - Escape character is '^']'.

SSH

- Herhangi bir parola girmeden uzak makineye ssh bağlantısı kurmayı sağlar
 - İstemcinin açık anahtarını (/etc/ssh/ssh_host_rsa_key.pub) sunucudaki (hedef) kullanıcıyı ev dizinindeki ~/.ssh/known_hosts dosyasına başına host ismi ekleyerek koyun.
 - Örneğin: İstemci açık anahtarı şu ise:
 - ssh-rsa AAAB3N.....kq8=
 - Hedef sunucuda /home/<kullanıcı>/.ssh/known_hosts dosyasına
 - istemci.host.name ssh-rsa AAAB3N.....kq8=

DHCP

- DHCP, bir yerel ağdaki istemcilere hızlı biçimde IP adresi dağıtmak ve böylece IP adresi çakışmalarının önüne geçmeye yarayan bir protokoldür.
- Linux üzerinde DHCP sunucusu kurmak oldukça kolaydır.
 - # apt-get install dhcp3-server



```
bor@bacula-test: ~  
File Edit View Terminal Help  
root@bacula-test:/etc/bind# apt-get install dhcp3-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  isc-dhcp-server  
Suggested packages:  
  isc-dhcp-server-ldap  
The following NEW packages will be installed:  
  dhcp3-server isc-dhcp-server  
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.  
Need to get 0 B/403 kB of archives.  
After this operation, 926 kB of additional disk space will be used.  
Do you want to continue [Y/n]? Y
```

DHCP

- DHCP ayar dosyası `/etc/dhcp/dhcp3.conf` dosyasıdır.
- Ayarlar yapıldıktan sonra DHCP sunucusu yeniden başlatılmalıdır.
`# /etc/init.d/dhcp3-server restart`
- Buna ek olarak, DHCP sunucusunun IP adresi dağıtacağı ağ bağlantısına dair ek ayarlar yapılmalıdır.
 - Bu amaçla `/etc/network/interfaces` dosyası düzenlenir.
 - Ağ servisi, `«/etc/init.d/networking restart»` komutu ile yeniden başlatılır.

DNS

- Uzun IP adreslerinin insanlar tarafından akılda tutulması zorlaşınca isimleri IP adresine, IP adreslerini isimlere dönüştüren DNS (Alan Adı Sunucusu) geliştirilmiştir.
- DNS hizmetini sunan en yaygın yazılım bind'dir.
- Bununla birlikte openDNS gibi alternatif yazılımlar da önemli bir yaygınlığa sahiptir.

\$host www.pardus.org.tr

www.pardus.org.tr is an
alias for
ns1.pardus.org.tr.

ns1.pardus.org.tr has
address
193.140.100.210

DNS

- DNS dağıtık bir veritabanı mimarisine sahiptir.
- Uç sistemler bilgilerini güncelleyebilmektedir.
- En üstten en alta kadar maksimum 127 seviye bulunabilir.

DNS

- Makinenin ismine bağlantı isteği olur.
 - İstemcinin bağlı olduğu DNS sunucusuna istenilen makinenin IP adresi sorulur.
 - DNS sunucusu IP adresini biliyorsa isteğe cevap verir, bilmiyor ise bir üst DNS sunucusuna sorar.
 - Bir üst DNS sunucusundan gelen IP adresi istemciye iletilir.
- Kök sunuculara fazla yük yaratmamak için kök'e sorulup öğrenilen DNS kaydı “Time to Live” (TTL) kadar süre DNS sunucunun önbelleğinde tutulur.
 - TTL süresinin çok uzun tutulması gereksiz istekleri engelleyerek ağın verimini arttırır, ancak DNS kayıtlarının güncellenmesini engellediğinden istenilen bilgiye ulaşmayı zorlaştırabilir.

DNS

- Ana DNS Sunucusu (Primary Master)
 - DNS kayıtlarını tutar.
 - DNS güncellemelerini yapar.
- Yardımcı DNS Sunucusu (Secondary Master)
 - DNS kayıtlarını kendisi tutmaz,
 - Birincil sunucudan alır.

DNS

- Linux İsim Çözme altsistemi
- /etc/resolv.conf:
 - Nameserver 144.122.199.20
 - Nameserver 212.175.1.4
 - search portakalteknoloji.com
- /etc/nsswitch.conf:
 - hosts: files dns
- /etc/hosts:
 - 127.0.0.1 localhost
 - 192.168.0.210 deneme.portakalteknoloji.com deneme

DNS

- Host komutu

```
$ host -t mx portakalteknoloji.com
```

```
$ host -t ns portakalteknoloji.com
```

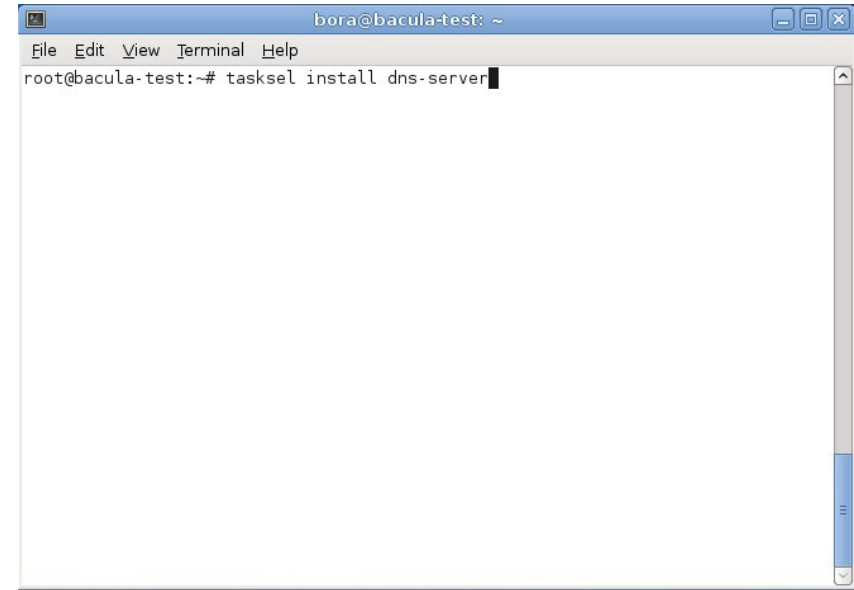
```
$ host deneme.portakalteknoloji.com
```

```
$ host deneme.portakalteknoloji.com ns.ISP.com
```

```
$ host -t any portakalteknoloji.com
```

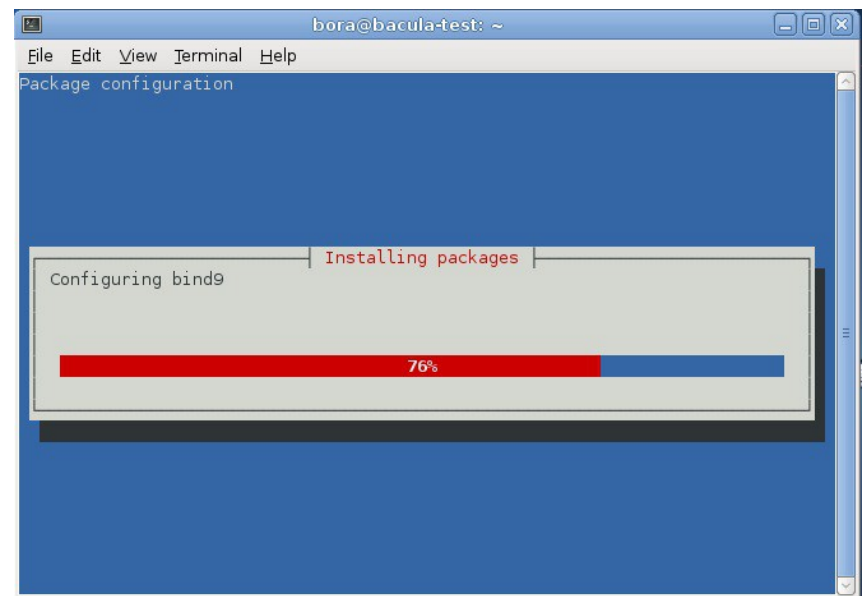
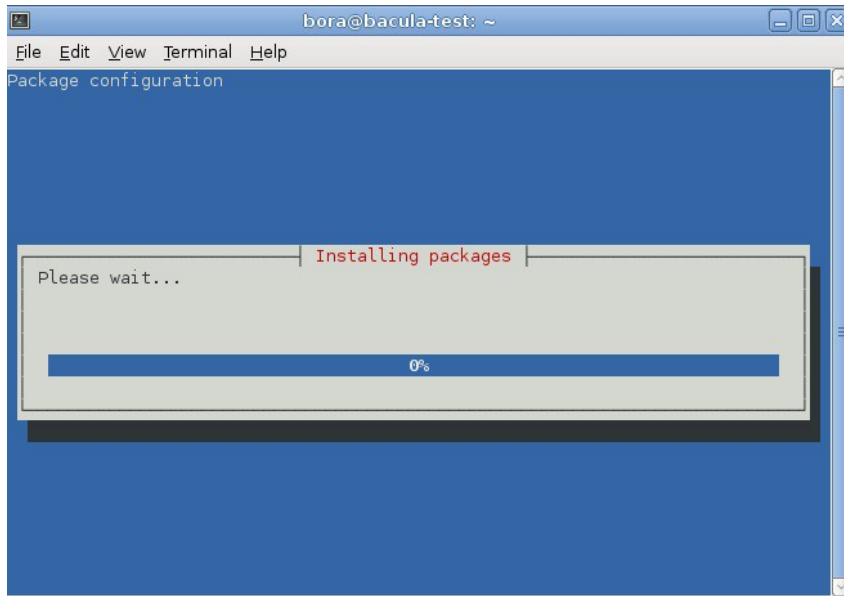
DNS

- Debian temelli sistemlerde DNS sunucu kurmak için «dns-server» isimli görevi, tasksel aracı ile kurmak yeterlidir.
 - Bu görev bind ve dnstools başta olmak üzere gereken tüm paketleri otomatik olarak kuracaktır.



```
bora@bacula-test: ~  
File Edit View Terminal Help  
root@bacula-test:~# tasksel install dns-server
```

DNS



DNS

- Bind ayarları /etc/bind dizini altında bulunan dört dosyada yer alır.
 - /etc/bind/named.conf – Ana ayar dosyasıdır ve diğer üç dosyadaki ayarların yüklenmesini sağlar
 - /etc/bind/named.conf.default-zones – Kök DNS sunucuları ve sunucunun DNS protokolü kapsamındaki davranışını ayarlar.
 - /etc/bind/named.conf.local – Yerel (sunucu) için geçerli ek ayarlar burada yapılır.
 - /etc/bind/named.conf.options – Çeşitli seçenekler (özellikle forwarder seçeneği) için ayarlar bu dosyada yapılır.

DNS

- DNS sunucu ile yapılabilecek ilgi çekici ayarlardan birisi, yerel ağdaki tüm istemcilerin <http://www.opendns.com/> sitesinden alınacak DNS ayarlarına uymasını sağlamaktır.
- Bu site, DNS temelli filtrelerden kaçınmanın yanı sıra, kurumların kendi ihtiyaçlarına uygun DNS filtrelerini devreye sokmalarını da sağlamaktadır.
- `/etc/resolv.conf` dosyasında

```
nameserver 127.0.0.1
```
- `/etc/bind/named.conf.options` dosyasında;

```
options {  
  forwarders { 208.67.222.222;  
                208.67.220.220; };  
  ...  
};
```
- *Arkasından bind yeniden başlatılacaktır:*

```
#!/etc/init.d/bind9 restart
```

NTP

- Yerel ağda çalışacak bir zaman sunucusu pratik bazı kolaylıklar sağlayacaktır.
- Kurulumu son derece kolaydır.

```
# apt-get install ntp
```

- Kurulumdan sonra /etc/ntp.conf ayar dosyası düzenlenmelidir.
- Ayarların ardından
 - NTP eşzamanlılığı sağlanmalıdır.
 - NTP sunucusu başlatılmalıdır.
 - NTP eşzamanlılığı doğrulanmalıdır.

NFS

- NFS, UNIX'deki geleneksel dizin paylaşım aracıdır.
- Uzak bilgisayardaki (sunucu) bir dizinin yerel bilgisayarda (istemci) dosya sistemine bağlanması esasına dayanır.
- Sistem son derece basit bir yapıya sahiptir. Ancak basit yapısı nedeni ile bazı konularda hassas ve kırılabilir olabilir.
 - NFS istemcilerinde eksik uygulama kurulumları,
 - Aynı kullanıcı adı için farklı bilgisayarlarda farklı kullanıcı tanımlayıcısı (uid) bulunması,
 - NFS trafiğinin güvenlik duvarından geçmesi,
 - DNS protokolüne olan bağımlılık
 - Zaman aşırımları

NFS

NFS Sunucusu

- Yazılım kurulumu
 - nfs-common
 - nfs-kernel-server
- NFS çekirdek modülünün yüklenmesi
- Gerekli ise paylaşım izinlerinin oluşturulması
- Sunucu ayar dosyası olan /etc/exports dosyasının (İng. «export file») hazırlanması
- NFS sunucusunun yeniden başlatılması

NFS İstemcisi

- Yazılım kurulumu
 - nfs-common paketi
- Açılışta NFS paylaşımının otomatik bağlanmasının sağlanması

NFS

- Örnek uygulamada sunucu üzerinde 192.168.0.2 adresinde /home/nfs dizinini sunulacaktır.
- NFS yazılım bileşenleri sistemi kontrol ederek tek tek kurulacaktır.
 - # apt-get install nfs-common
 - # apt-get install nfs-kernel-server
- Kurulumlar sonunda NFS servisi kontrol etmek için araçlar
 - # rcpinfo -p | grep nfs
 - # cat /proc/filesystems | grep nfs
 - # rcpinfo -p | grep portmap
- Eğer çıktılarda NFS ile ilgili çıktı yoksa, NFS çekirdek modülü yüklenmemiştir. Elle yüklemek gerekir.
 - #modprobe nfs

NFS

- NFS sunucusundaki paylaşım dizini /home/nfs olacaktır.
- Eğer bu dizin oluşturulmamış ise yaratılması gerekir.
- Ayarlar /etc/exports dosyasında yapılacaktır.
 - <erişilecek dizin> <erişmesine izin verilen alt ağ> (<seçenekler>)
- Örnekler
 - /home/nfs 192.168.0.20(rw,sync) – Belirtilen IP adresine yazma okuma ve eşzamanlı çalışma kipinde bağlantı izni ver.
 - /home/nfs 192.168.0.0/24(ro,sync) – Belirtilen IP bloğundaki istemcilere salt okuma ve eşzamanlı çalışma kipinde bağlantı izni ver.
 - /home/nfs 192.168.0.20(rw,sync) 192.168.0.0/24(ro,sync) – Önceki iki ayarın birlikte yazılmış hali
 - /home/nfs *(ro,sync) – Dizine herhangi bir IP adresinden salt okuma ve eşzamanlı kipte bağlantı izni ver.
 - /home/nfs *.meb.gov.tr(ro,sync) – Dizine alan adı .meb.gov.tr olarak ayarlanmış tüm bilgisayarlardan salt okuma kipinde ve eşzamanlı bağlantı izni ver.

NFS

- Test amaçlı olarak NFS ayar dosyasında yapılacak olan ayar 192.168.0.0/24 bloğuna salt okunur erişim hakkı vermek olacaktır.

```
# echo '/home/nfs/ *(ro,sync)' > /etc/exports
```

- Test etmek için

```
# tail -1 /etc/exports
```

- Bir adet deneme dosyasını oluşturmak için

```
# touch /home/nfs/nfs-test-dosyasi
```

- NFS sunucusunu yeniden başlatmak gerekecektir.

```
# /etc/init.d/nfs-kernel-server restart
```


NFS

- İstemcide NFS alt yapısı kısmen kurulacaktır. Sunucu paketinin kurulmasına gerek yoktur.
apt-get install nfs-common
- İstemcide NFS olarak bağlanacak olan uzak dizinin, yerel bir dizine bağlanması gerekecektir. Örnekte /home/nfs_local dizini olarak adlandırılacaktır.
mkdir /home/nfs_local
mount -t nfs 192.168.0.2:/home/nfs /home/nfs_local
- Sunucu ve istemcinin NFS sürümleri farklı ise istemci kendi sürümünü belirtmelidir.
mount -o **nfsver=3** -t nfs 192.168.0.2:/home/nfs /home/nfs_local

NFS

- İstemcide elle bağlanan dizinin çalıştığından emin olunduktan sonra, bağlama işleminin otomatik hale getirilmesi işlem kolaylığı sağlanacaktır.
- Bu amaçla /etc/fstab dosyasına bir satır eklenecektir.

```
# echo "10.1.1.50:/home/nfs /home/nfs_local/ nfs defaults 0 0" >>  
/etc/fstab
```
- Kontrol etmek için

```
# tail -1 /etc/fstab
```

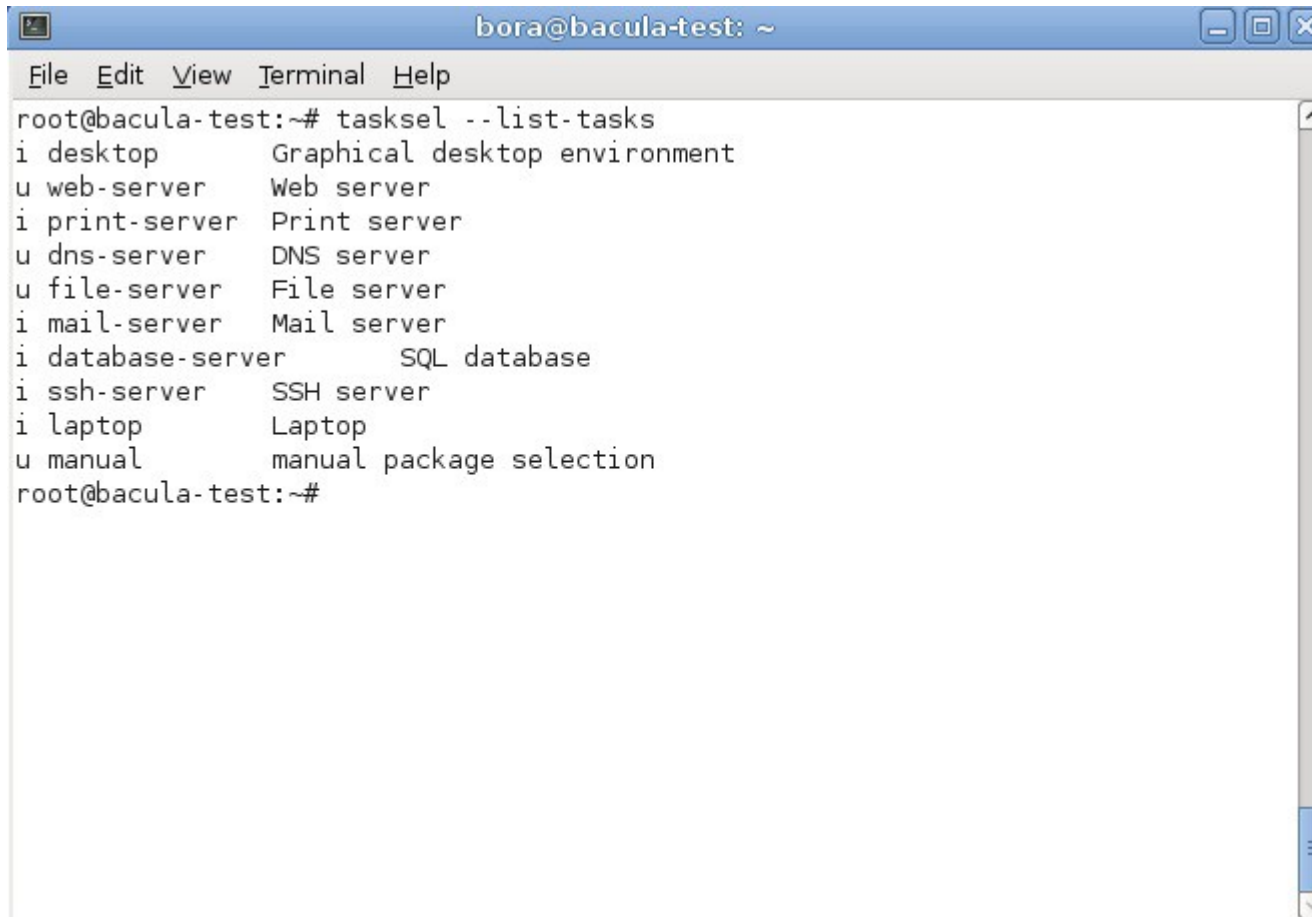
NFS

- Bir sistem, /home dizinini topluca NFS'den bağlayacak olursa o zaman kullanıcı ev dizinleri de otomatik olarak uzak sunucudan alınacaktır.
 - Bu durumda bağlanan dizindeki erişim hakları (750, 644, vs) uzak sunucudaki kullanıcı tanımlayıcıları (İng. User ID) için oluşturulmuş olacaktır.
 - Dolayısı ile uzak sistem ile yerel sistemde aynı adlı kullanıcının kullanıcı adı ve tanımlayıcısının aynı biçimde eşleşmesi gerekir.
- Bunu sağlamanın iki yolu vardır:
 - Her iki sistemde kullanıcıların aynı şekilde yaratılması Örneğin, kullanıcıların tek tek yaratılması yerine ortak bir betiğin işletilmesi olabilir.
 - İstemcilerin kullanıcı adı ve parola denetimlerini de merkezi bir biçimde yapması.

SMB/CIFS

- Samba, *nix işletim sistemleri ile Windows platformu arasında iletişimi sağlayan bir ağ sunucusudur.
 - Samba,
 - Windows makinenin dizinlerini UNIX ile paylaşır.
 - Windows ağlarına alan denetleyicisi olabilir.
 - UNIX'in disk bölümünü Windows ile paylaşır.
 - Windows yazıcısını UNIX'e paylaşır ya da tam tersini yapabilir.
- Samba ile dosya paylaşımı sunucu alt yapısı, Debian temelli sistemlerde tasksel ile kurulabilen «file-server» görevi ile sağlanır.

SMB/CIFS

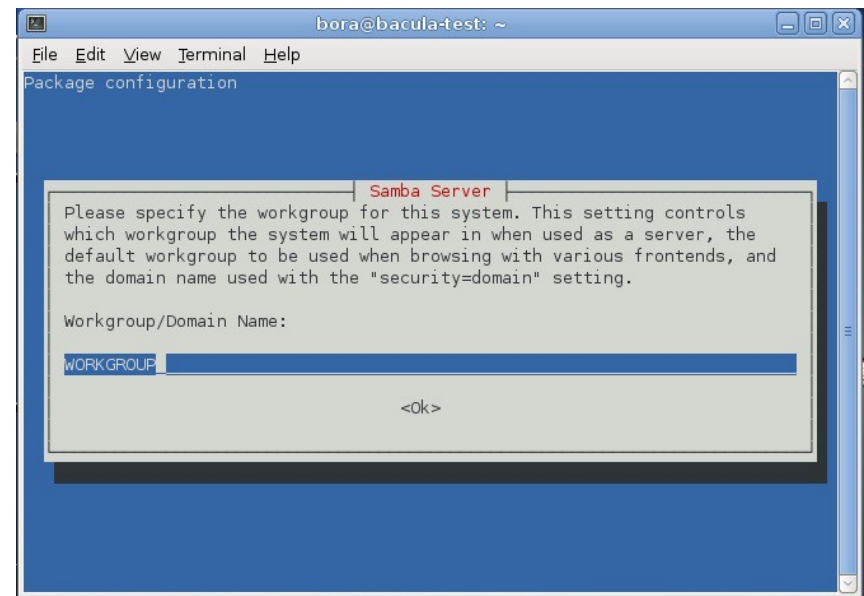
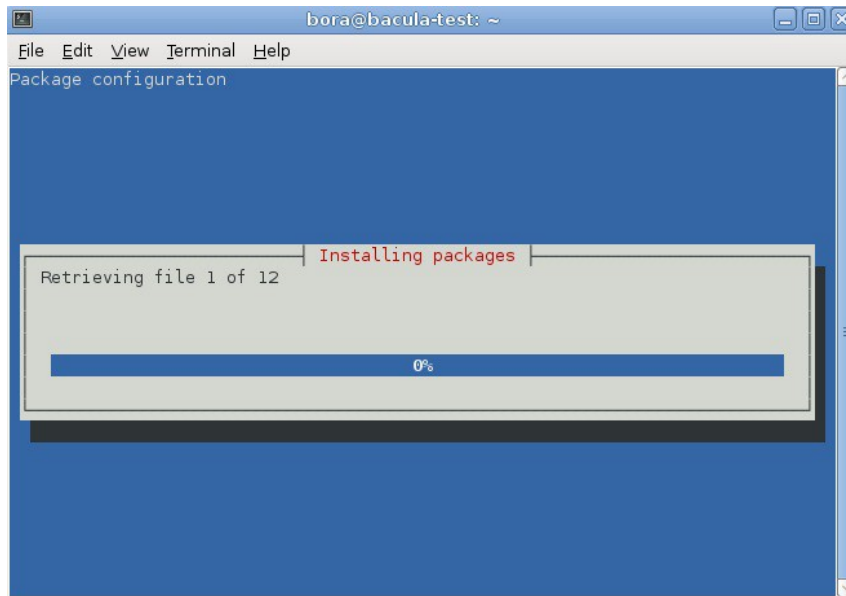


```
bora@bacula-test: ~  
File Edit View Terminal Help  
root@bacula-test:~# tasksel --list-tasks  
i desktop      Graphical desktop environment  
u web-server   Web server  
i print-server Print server  
u dns-server   DNS server  
u file-server  File server  
i mail-server  Mail server  
i database-server      SQL database  
i ssh-server   SSH server  
i laptop      Laptop  
u manual      manual package selection  
root@bacula-test:~#
```

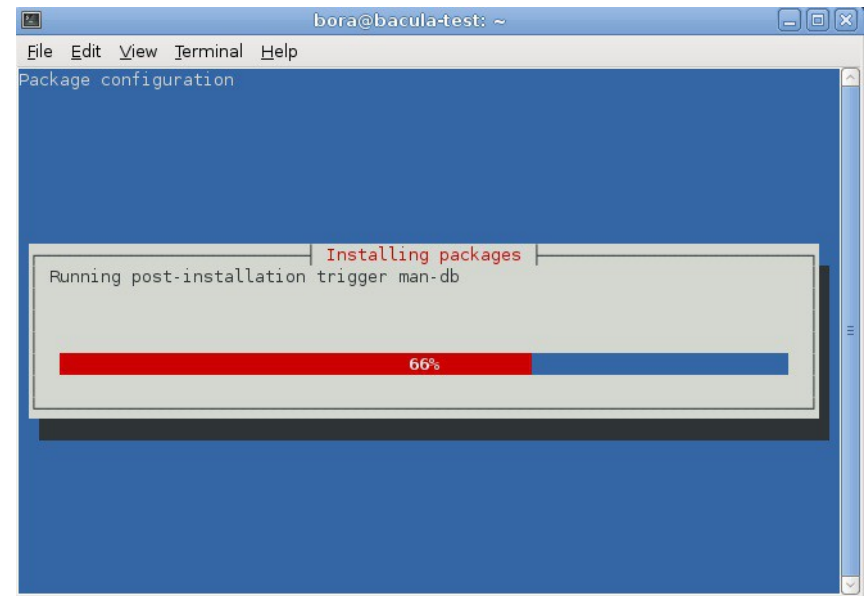
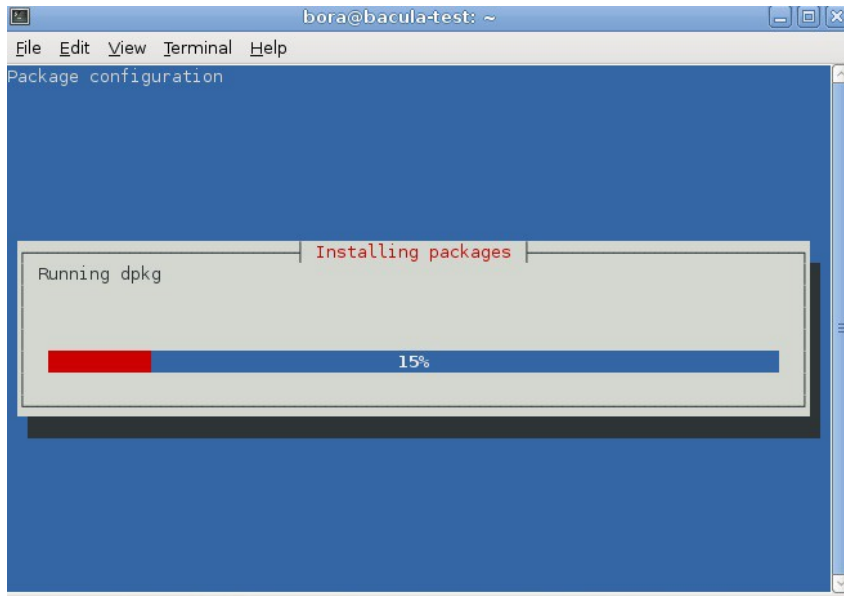
SMB/CIFS

- Samba arkaplanında iki sunucu çalışır.
 - `smbd`: Yazıcı ve dosya paylaşımı sunucusudur.
 - `nmbd`: WINS istemcisi/sunucusudur.
- Samba'da istemci olarak ise, dört bileşen bulunur.
 - `smbclient`: Linux'ta Dosya Paylaşım istemcisidir.
 - `smbmount`: İstemcilerin paylaşımlarını bağlamak için kullanılır.
 - `smbumount`: Dosya paylaşımlarını ayırmak için kullanılır.
 - `smbpasswd`: SMB parolası belirlemek için kullanılır.
- Kurulum için
 - # `tasksel install file-server`**
 - Debian kurulum DVD'si gereken paketleri içerdiği için ağ bağlantısı gerekli olmayacaktır. Ancak güncelleme için ağ bağlantısı gerekir.

SMB/CIFS

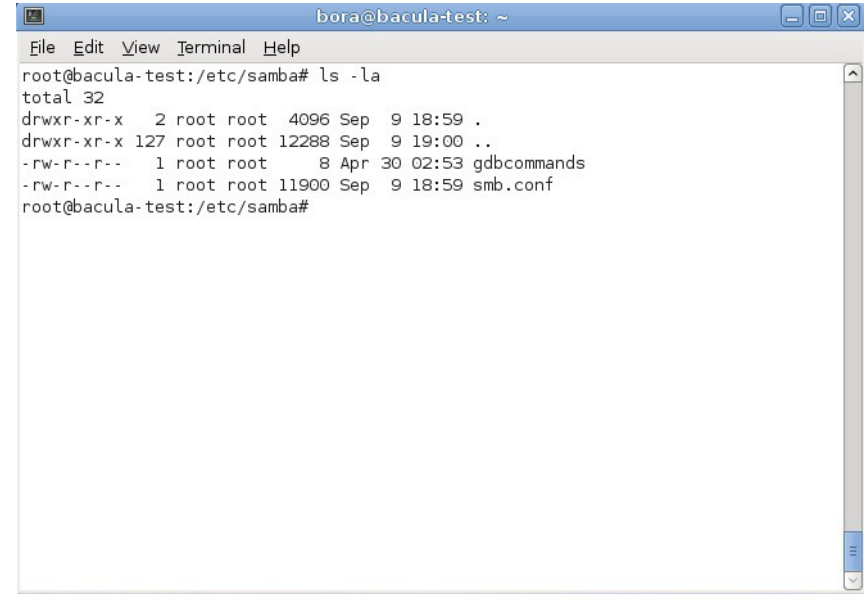


SMB/CIFS



SMB/CIFS

- Samba ayar dosyası «/etc/samba/smb.conf» dosyasıdır.
 - [global]: Samba'nın genel ayarları
 - [homes]: Ev dizinlerinin paylaşımları
 - [netlogon]: Alan oturumları için.
 - [Profiles]: Özel dolaşım profili için
 - [printers]: Ağda bulunan yazıcıların paylaşımları
 - [tmp]: Paylaşım için geçici dizinler
 - [public]: Kamuya açık paylaşım
- Man sayfalarında kapsamlı bilgiler vardır.



```
bora@bacula-test: ~  
File Edit View Terminal Help  
root@bacula-test:/etc/samba# ls -la  
total 32  
drwxr-xr-x  2 root root  4096 Sep  9 18:59 .  
drwxr-xr-x 127 root root 12288 Sep  9 19:00 ..  
-rw-r--r--  1 root root    8 Apr 30 02:53 gdbcommands  
-rw-r--r--  1 root root 11900 Sep  9 18:59 smb.conf  
root@bacula-test:/etc/samba#
```

SMB/CIFS

[global]

log file = /var/log/samba/%m.log //Tutulacak olan gunlugun yeri , %m
NetBIOS ismini ekler.

load printers = yes //printcap dosyasında tanımlı yazıcılar paylaşımına
açık

idmap gid = 16777216-33554431

socket options = TCP_NODELAY SO_RCVBUF=8192
SO_SNDBUF=8192

winbind use default domain = no

template shell = /bin/false

dns proxy = no

server string = Samba Server //Agdaki aciklama cumlesidir.

SMB/CIFS

```
// [global] devam
password server = None
idmap uid = 16777216-33554431
path = /home/bora
default = homes
workgroup = WORKGROUP //samba sunucunun bulunduğu alan adı
os level = 20 //samba çevre sistemlere bulundugunu bildirecektir. Bu
deger onun seviyesidir.
printcap name = /etc/printcap
security = share
max log size = 50
```

SMB/CIFS

NOTE: If you have a BSD-style print system there is no need to
specifically define each individual printer

[printers]

comment = All Printers

path = /var/spool/samba //yazıcı kuyruk dizini.

browseable = no

Set public = yes to allow user 'guest account' to print

printable = yes

public = yes

SMB/CIFS

Sisteme kullanıcı_adi/parola ile oturum açan bütün kullanıcıların ev dizinleri kullanıcı adları ile paylaşılacak.

[homes]

comment = Home Directories

browseable = yes

writable = no //Yazılamaz

readonly = yes // Sadece okunabilir.

SMB/CIFS

- Samba'da ağ üzerinden paylaşımlara erişmek için smbclient, erişilen paylaşımları yerel bir dizine bağlamak için ise smbmount komutları kullanılır.

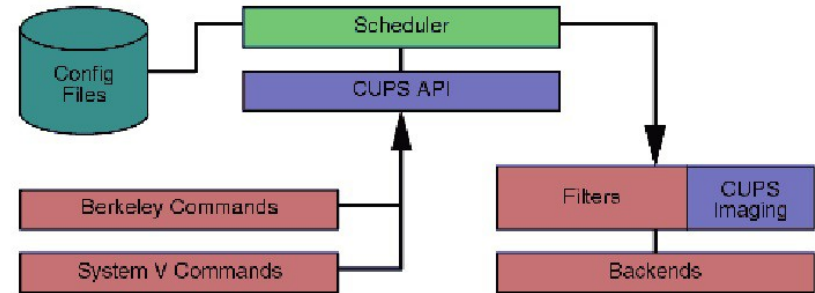
```
smbclient //makine/<dizin> -U <kullanıcı>
```

```
smbclient -L //makine
```

```
smbmount //makine/<dizin> <yerel_dizin> -o  
username="<kullanıcı>"
```

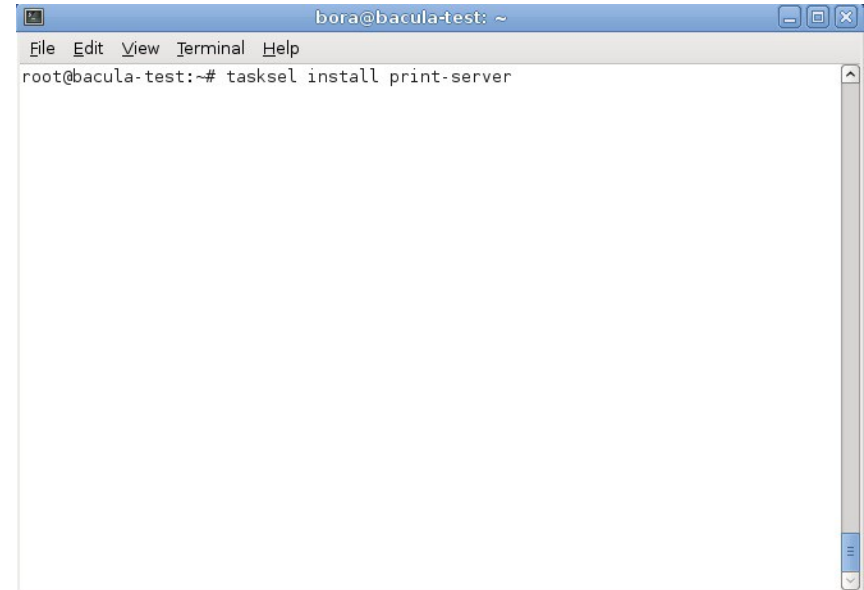
CUPS

- Linux dağıtımlarında yazıcıların yönetilmesi için kullanılan alt yapının adı CUPS'dır.
- CUPS UNIX temelli bir alt yapıdır ve bu nedenle aynı zamanda yazıcıların ağ üzerinde paylaşılmasını da sağlar.



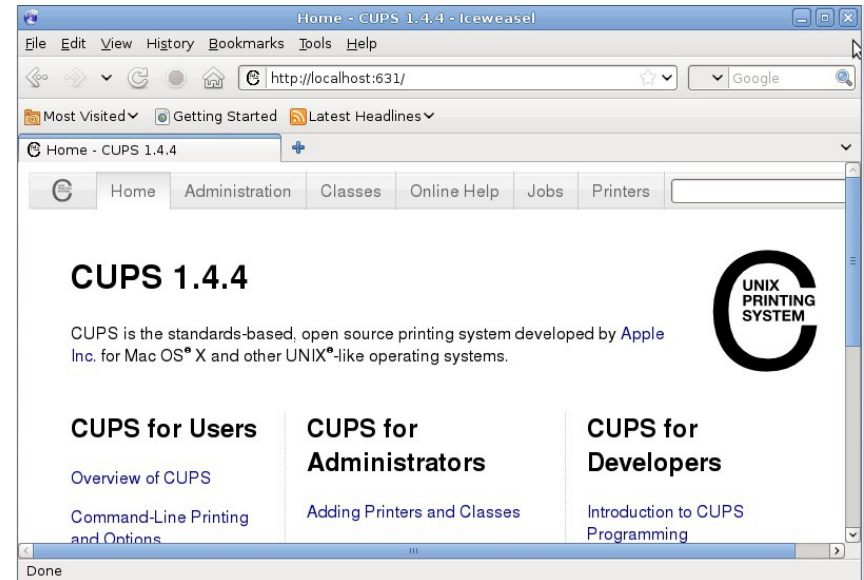
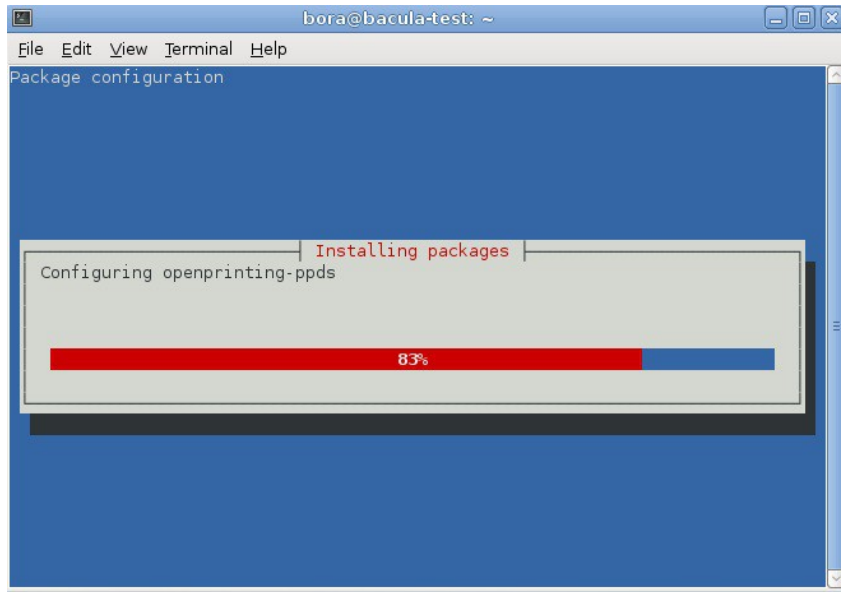
CUPS

- CUPS, Debian temelli sistemlerde «desktop» ve «print-server» adlı görevlerin bir parçasıdır.
 - Dolayısı ile herhangi bir masaüstü yöneticisi kurulurken CUPS otomatik olarak kurulur.
 - Ancak masa üstü yöneticisi olmayan bir bilgisayarı da kolayca yazıcı sunucusu yapmak mümkündür.
- # tasksel install print-server**



```
bora@bacula-test: ~  
File Edit View Terminal Help  
root@bacula-test:~# tasksel install print-server
```


CUPS



CUPS

- Baskı işlerinin yönetimi için ise iki komut kullanılır
 - lpq – Yazıcıda bekletilen işleri gösterir
 - cancel – Numarası verilen yazıcı işi iptal edilir.
- CUPS'ın yönetilmesinde web arayüzü yoğun biçimde kullanılır.
 - `http://<sunucu_adresi>:631/`
- Ayrıca `/etc/cups/cupsd.conf` dosyası da düzenlenebilir.

VNC

- VNC (Virtual Network Computing) yaygın bir uzak masaüstü protokolüdür.
 - Ekran kartının tampon belleği paylaşmak yolu ile yüksek performanslı bir uzak masaüstü deneyimi sunar. Bu teknik işletim sisteminden bağımsız olduğu için tüm işletim sistemlerinde VNC kullanmak mümkündür. Hatta Java için dahi VNC API'leri vardır.
 - VNC orijinal olarak Olivetti ve Oracle tarafından geliştirilmiştir. Daha sonra AT&T tarafından da ekler yapılmıştır.
 - Orijinal sistem günümüzde gelen VNC günümüzde bir çok eklenti içermektedir.
 - Ses
 - Dosya aktarımı
 - Kopyala yapıştır (Latin-1)

VNC

- VNC yerel ağda oldukça verimlidir. İnternet üzerinden de erişim sorunsuz yapılabilir ancak ağ ayarları (NAT, güvenlik duvarları, vb) konusunda dikkat etmek gereklidir.
- VNC yerel ağ için tasarlandığından dolayı kendisine ait bir güvenlik özelliği yoktur. Bu nedenle İnternet üzerinden uzak masaüstüne bağlantı için SSH tüneli veya VPN türü ek güvenlik önlemleri alınması önerilir.
- Yaygın VNC uygulamaları
 - TightVNC
 - TurboVNC
 - TeamViewer (ticari)

RDP

- RDP (Remote Desktop Protocol) Microsoft tarafından geliştirilen bir uzak masaüstü protokolüdür.
 - Protokolün üstünlükleri yüksek renk derinlikleri, ses, yazıcı yönlendirmesi, dosya sistemi yönlendirmesi, ağ üzerinden kullanıcı doğrulama gibi ek kabiliyetlerdedir.
 - Windows içinde RDP sunucu ve istemcileri entegre gelmektedir. Linux için ise kurulumu gereklidir.
- RDP güvenlik açısından çeşitli riskler içerir. SSL/TLS kullanımı bu riskleri kısmen gidermektedir. Özellikle RDP bağlantıları üzerinden parola çalınması olasılığı daima mevcuttur.
- Yaygın Linux RDP yazılımları
 - Rdesktop
 - FreeRDP
 - KRDC

5. Bölüm

Sanallaştırma

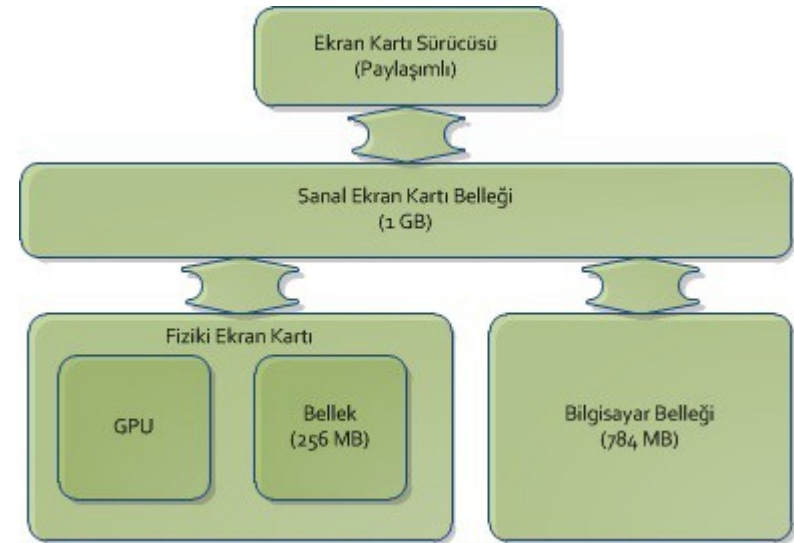
- Bu bölümde, sanallaştırmadan ve yaygın kullanılan VirtualBox aracından bahsedilecektir.
- Sanallaştırmanın değişik uygulama alanları olabilir. Ancak burada daha çok gündelik kullanım amaçlarına odaklanılacaktır.

Sanallaştırma

- Sanallaştırma, bir varlığın kullanıcısına sağlanması sırasında gerçek kaynağının kullanıcı için önemsiz olacağı derecede saklanması kavramına verilen isimdir.
- İnternet çok tipik bir örnektir. Google'da arama yaparken, sorgunuzun binlerce Google sunucusundan hangisi tarafından işlendiğini önemsemezsiniz. Sizin için önemli olan sonucun gelmesidir.
- Aynı kavramı küçük ölçekte uygularsanız sanal makineler ile karşılaşabilirsiniz.
- Bir işletim sistemi için, uygun alt yapı sağlanırsa, hangi fiziki donanımı kullandığı önemsiz hale getirilebilir.
 - İşlemci
 - Bellek
 - Disk
 - Kartlar

Sanallaştırma

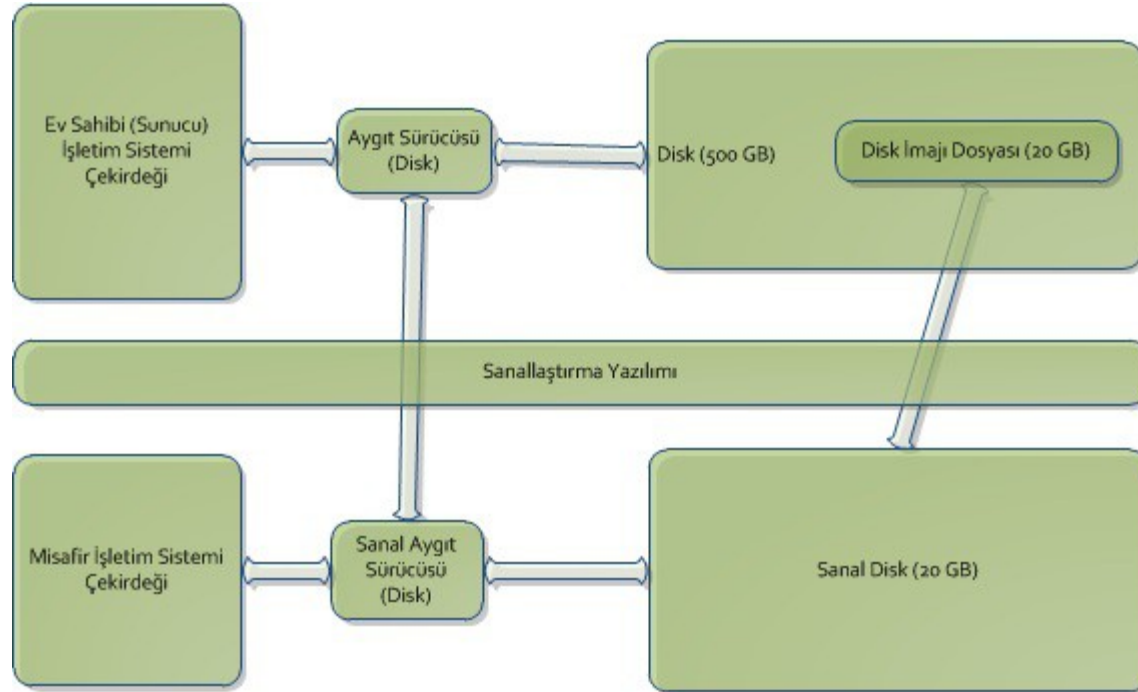
- Bir işletim sistemi için tüm fiziki donanımlar aslında aygıt sürücüleri tarafından erişilen varlıklardır.
- Aygıt sürücüsüne sanal donanımlar tanıtmak mümkündür.
- “Paylaşımlı” bellekli ekran kartları zaten böyle çalışır.



Sanallaştırma

- Sanallaştırılan bir işletim sistemi için de bilgisayara ait tüm kaynaklar paylaşılmaktadır.
 - İşlemci
 - Bellek
 - Disk
 - Ekran
 - CD/DVD sürücüsü
 - USB aygıtlar
 - Ağ Bağlantısı

Sanallaştırma



Sanallaştırma

- Linux, işletim sistemi sanallaştırması alanında öncü bir işletim sistemidir.
- Ticari sanallaştırma alt yapıları dahil hemen tüm sanallaştırma çözümleri Linux için geliştirilen sanallaştırma kodlarından yararlanır.
- Temel bileşenler
 - Qemu
 - Xen
 - KVM
- Yaygın ve kolay kullanılır bir araç olarak Oracle/SUN VirtualBox Pardus'da en çok kullandığımız sanallaştırma aracıdır.

Sanallaştırma

- Öncelikle, hangi amaçla sanallaştırma yapacağımıza karar verilmesi gereklidir.
 - Deneme-Yanılma
 - Kaynakların izlenmesi
 - Lisans Yönetimi
 - Uzaktan Erişim
 - Güvenlik
 - Konsolidasyon

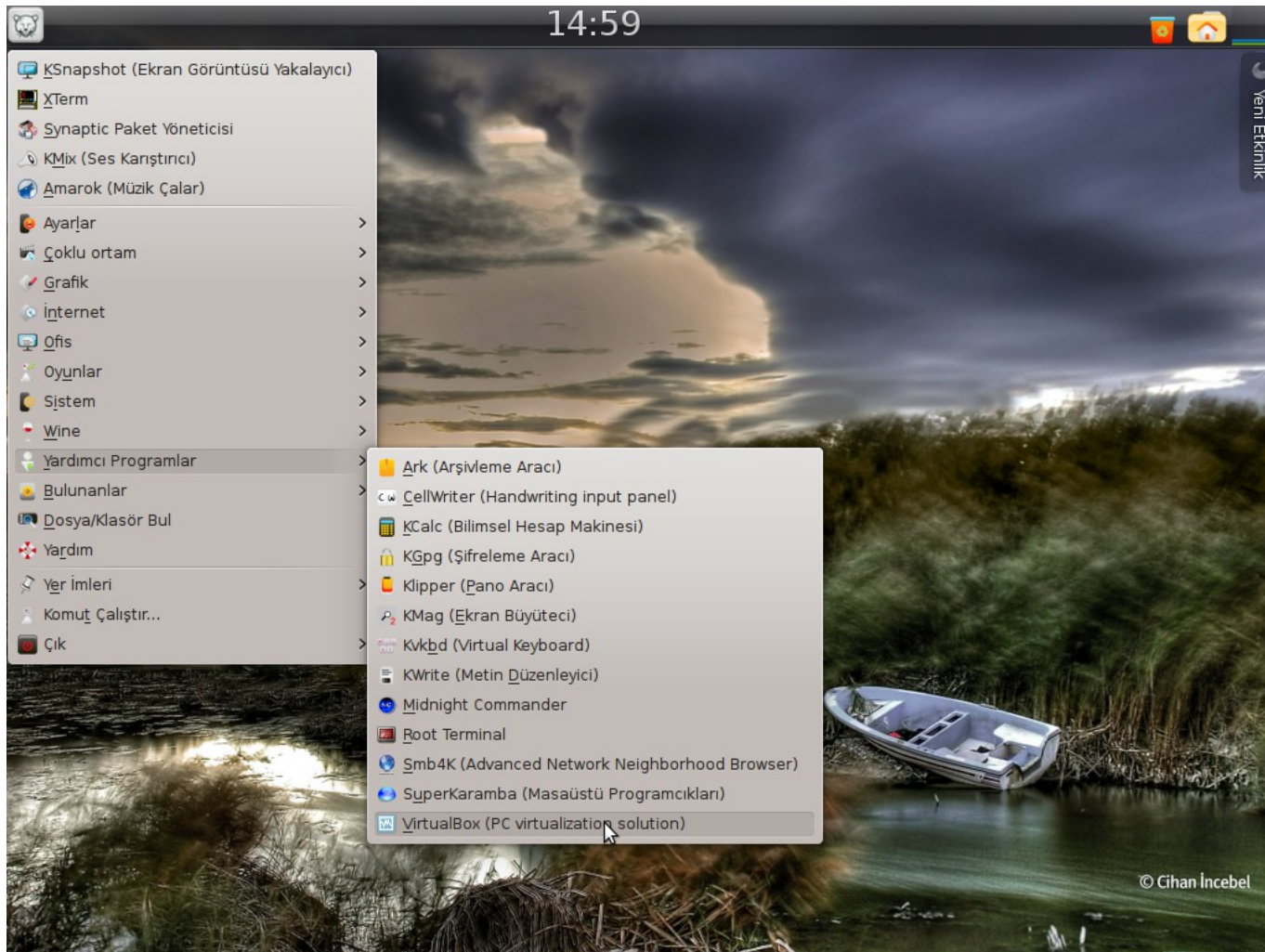
Sanallaştırma

- Ardından, sanal makinemiz için planlama yapılmalıdır.
 - İşlemci kapasitesi
 - Bellek kapasitesi
 - Disk adedi ve türleri
 - Ağ bağlantısının nasıl yapılacağı
 - Güvenlik
 - İşletim sistemi
- Sanallaştırma yapılacak olan sistemin masaüstü veya sunucu olması da önemli bir parametredir.
 - Masaüstü sistemlerin sürekli olarak yüksek performans verecek şekilde çalışacağı öngörülmediği için donanım tasarımları ve üretimleri de buna göre yapılmaktadır.
 - Bazı işlemci modelleri ile gelen bilgisayarlarda sanallaştırma düşük verimli çalışacaktır.

VirtualBox

- Virtualbox'ın ana paketi olan virtualbox-ose üç ayrı paketin daha kurulmasını gerektirir.
 - libgsoap2 – C ve C++ üzerinden SOAP temelli web servislerinin oluşturulması için gerekli kitaplıklar.
 - libvncserver – Sanal makinelerin ekranlarını VNC üzerinden sunmak için gerekli kitaplıklar.
 - virtualbox – Virtualbox temel kitaplıkları.
- Grafik arayüz uygulamasının paketi virtualbox-ose-qt adlı pakettir.
 - Bu paketi de Synaptic üzerinden yüklemek mümkündür.
- Kurulum sonrasında, Virtualbox, ana menüde «Yardımcı Programlar» menüsü altında yer alacaktır. Buradan başlatmak mümkündür.

VirtualBox



VirtualBox

- Virtualbox'ın kurulumu sırasında zorunlu olmayan ancak kurulması önerilen bir dizi ek paket bulunmaktadır.
- Önerilen paketler
 - libgl1 ve libqt4-opengl – OpenGL kitaplıkları
 - virtualbox-ose-dkms – Çekirdek modülü kodları
 - libasound2 ve libpulse0 – Ses kitaplıkları
 - vde2 – Sanal dağıtık ethernet

VirtualBox

- Virtualbox QT arayüzü ilk kez açıldığı zaman, henüz bir sanal makine yaratılmamış olduğu için sanal makine listesi boş gelecektir.
- Araç çubuğundaki «Yeni» isimli düğmeye tıklayarak ilk sanal makinenin oluşturulması işlemlerine başlanabilir.



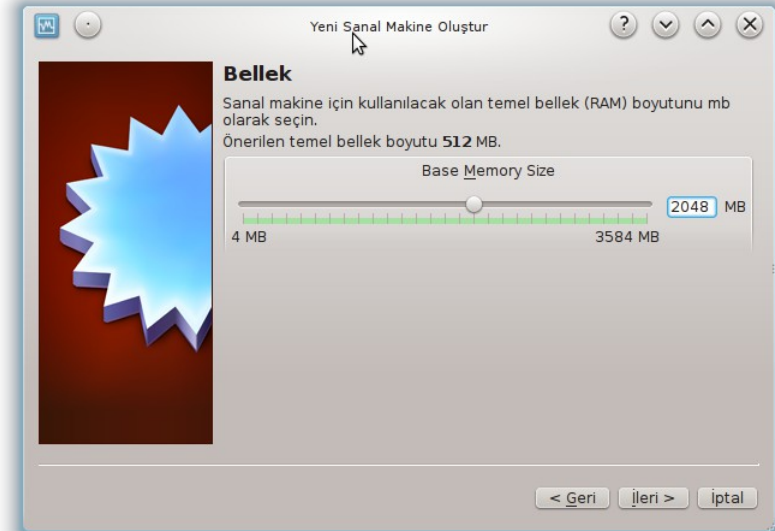
VirtualBox

- Sanal makine oluşturulması aslında sanal bir fiziki donanımın tasarlanması anlamına gelmektedir. Öte yandan, bir bilgisayarda aynı anda bir çok sanal makinenin çalışabileceği de asla gözardı edilmemeli, bu nedenle sanal makinelerin yönetiminin kendi başına bir iş kalemi olacağı düşünülmelidir.
- Yaratılan her sanal makinenin bir ismi olacaktır – ki bu isim diğerlerinden farklı olmalıdır.



VirtualBox

- Yarattılan sanal makinenin en önemli iki bileşeni sanal makineye tahsis edilen bellek ve disk alanları olacaktır.
 - Bellek miktarı belirlenirken şunlara dikkat etmek gerekir.
 - Ev sahibi işletim sistemi de bellek tüketecektir. Bu nedenle özellikle masaüstü barındıran ve gündelik amaçlarla kullanılan bir işletim sistemi söz konusu ise sistem belleğinin en az yarısının ev sahibine bırakılması sağlıklı olacaktır.
 - 32 Bit bir işletim sistemine 3,3 GB'dan fazla bellek vermenin pek anlamlı olmayacağı gözardı edilmemelidir.



VirtualBox

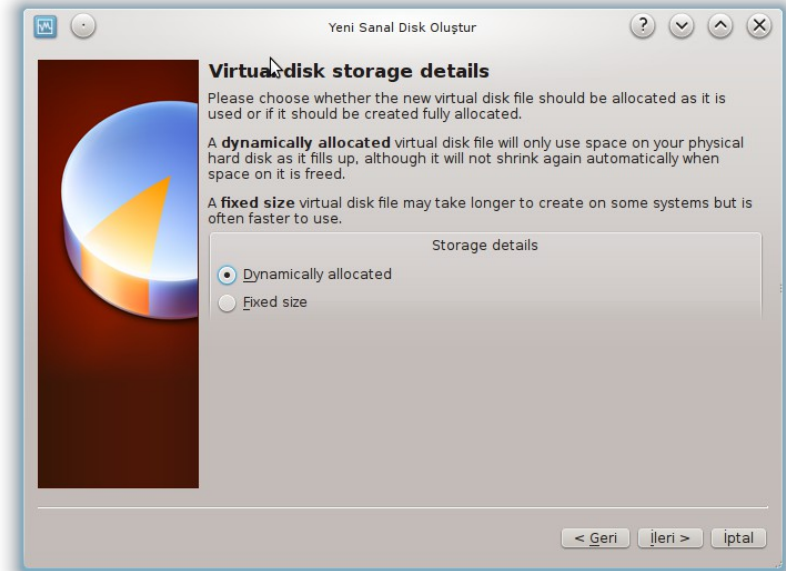
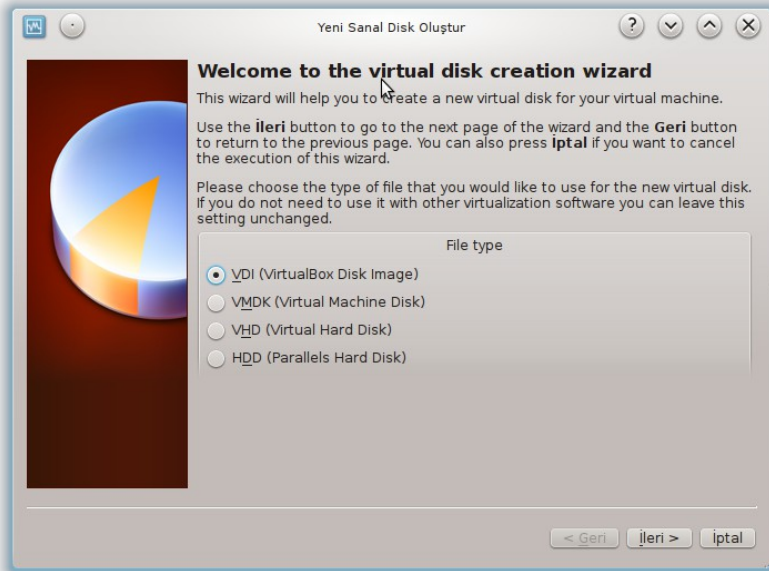
- Sanal makinenin kullanımı için oluşturulacak olan sanal diskle ilgili olarak bir kaç konuya dikkat edilmelidir.
 - Yaratılan diskin boyutları kurulacak olan işletim sisteminin ihtiyaçlarına yeterli olacak mıdır? İhtiyaçların takas alanı, kullanıcı dosyaları gibi ek alanları da içerecek şekilde değerlendirilmesi önemlidir. Virtualbox yazılımının önerdiği disk boyutları genellikle mutlak asgari değerlerdir ve bu değerlerin altına inmek doğru olmaz; ancak bu değerlerin biraz üzerine çıkılması da yararlı olacaktır.
 - Yaratılan diskin türü, genişleme modeli ve sanal diski temsil eden dosyanın bulunduğu disk bölümündeki dosya sistemi nedir? Bu özellikler diskin performansı kadar diskin farklı bir işletim sistemine taşınarak orada yeniden sanal makineye bağlanması konusundaki becerileri belirleyecektir.



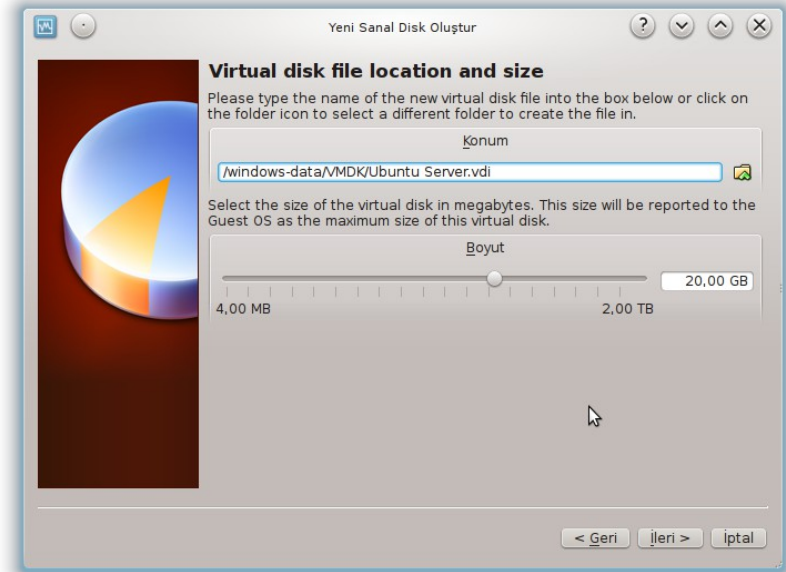
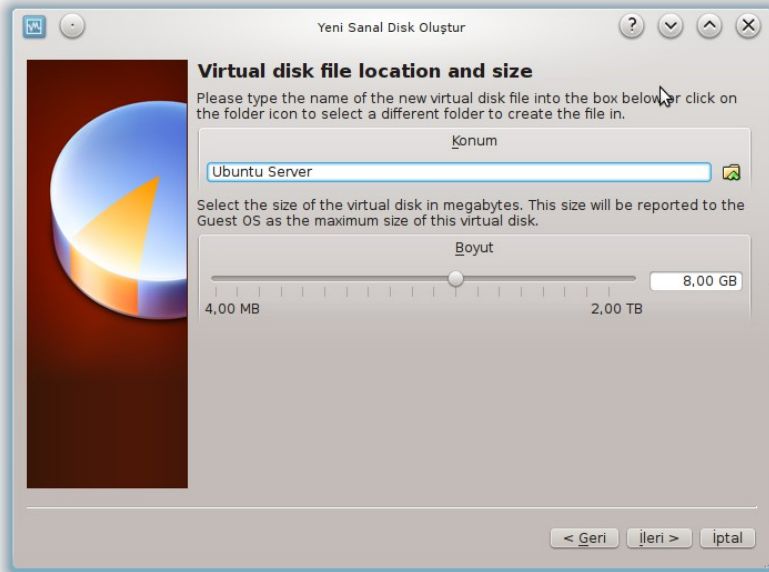
VirtualBox

- Virtualbox bir çok değişik biçimde disklerle çalışabilir. Ancak en yaygın kullanılan iki format VDI (kendi formatı) ile VMDK (VmWare formatı) olmaktadır.
- Oluşturulan sanal diskin genişleme modeli için iki farklı alternatif bulunmaktadır.
 - Dinamik genişleme – Diskin sanal boyutundan küçük bir yer kaplamasına izin verilir. Diske yazma gerçekleştirildiğinde diskin gerçekte kapladığı alan artar.
 - Sabit boy – Diskin sanal boyutu kadar alan derhal ve tek bir dosya olarak ayrılır. Böylece diske yazma gerçekleştirildiğinde diskin boyutu artmaz.
- Dinamik olarak genişleyen bir yapısı olması, pratik avantajlar sağlamakla birlikte, bazı dezavantajları da vardır:
 - Diski temsil eden dosya sürekli olarak büyümekte olduğu için bu dosyaya yazmalar biraz daha yavaş olur.
 - Diski temsil eden dosya küçük parçalar halinde genişlediği için dosyanın bulunduğu disk bölümündeki dosya sistemine bağlı olarak dosyanın disk üzerinde dağınık biçimde yerleşmesi (İng. Fragmentation) problemi yaşanabilir. Bu da yine diskten okuma ve diske yazma performansını olumsuz etkiler. Aynı sorun, diskin başka bir disk bölümüne kopyalanarak yedeklenmesi için geçen süreyi de artıracaktır.
 - Bu nedenle zorunlu olmadıkça sanal makinelerin disklerine ait dosyalar NTFS ve FAT32 dosya sistemlerinde bulundurulmaz.

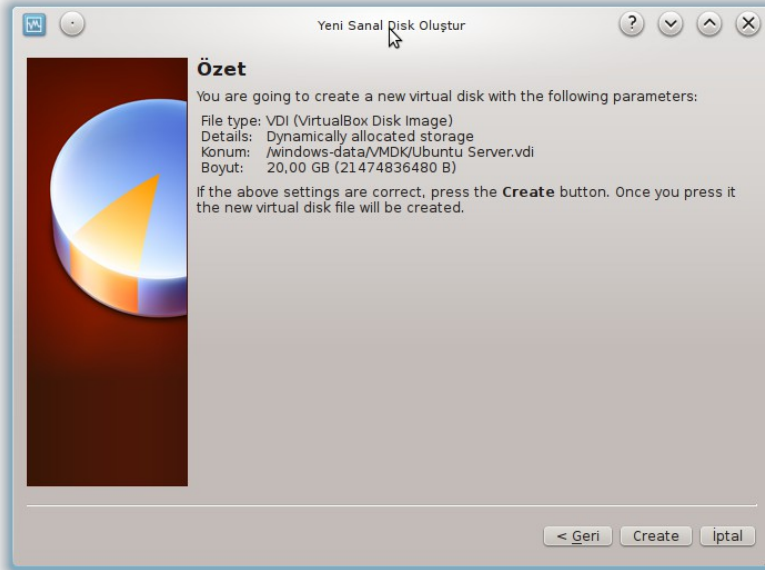
VirtualBox



VirtualBox



VirtualBox



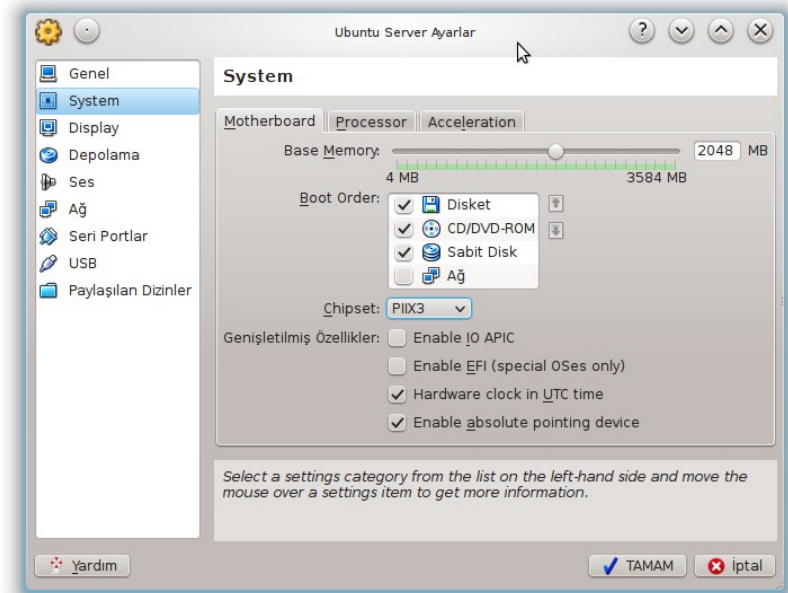
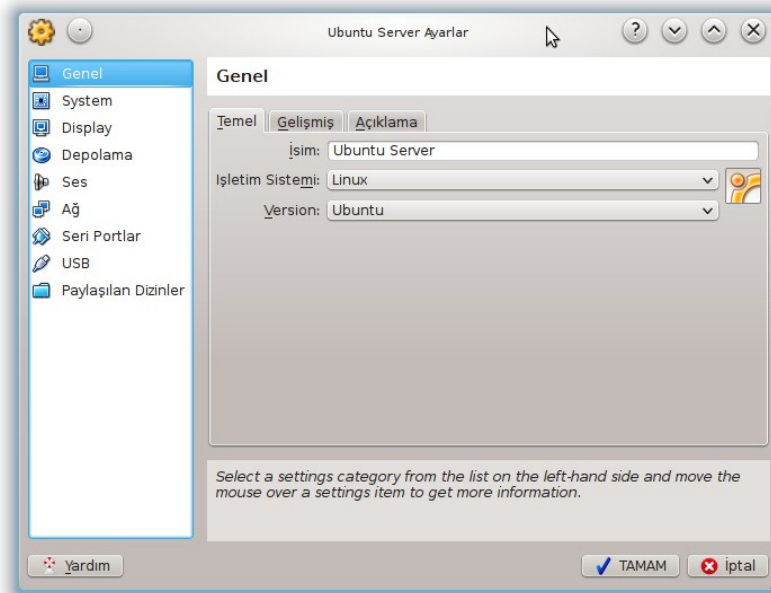
VirtualBox

The screenshot displays the Oracle VM VirtualBox Manager interface. The main window is titled "Oracle VM VirtualBox Manager" and shows the settings for a virtual machine named "Ubuntu Server". The interface is divided into several sections:

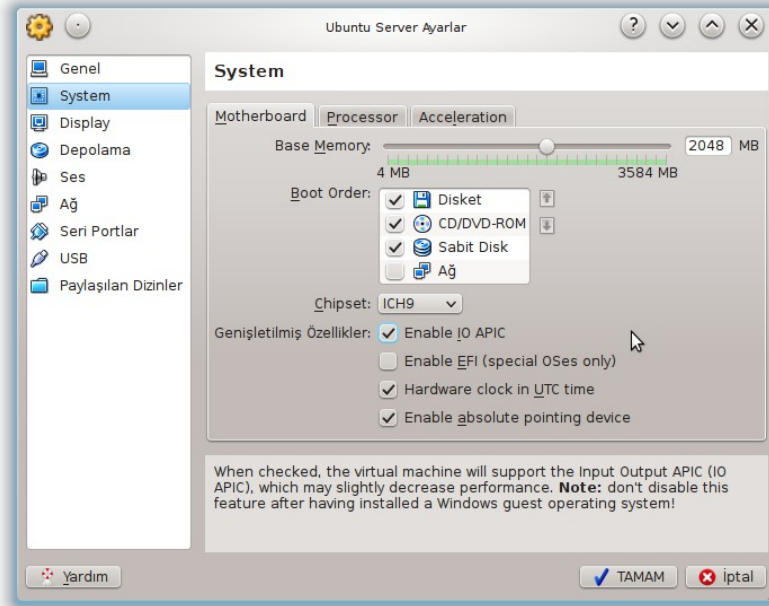
- General:** Name: Ubuntu Server, Operating System Type: Ubuntu.
- System:** Base Memory: 2048 MB (1 ?), Boot Order: Diskette, CD/DVD-ROM, Hard Disk, Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX.
- Display:** Video Card Memory: 12 MB (1 ?), Remote Desktop Server: Disabled.
- Storage:** IDE Controller, IDE Secondary Master (CD/DVD): Empty, SATA Controller, SATA Port 0: Ubuntu Server.vdi (Normal, 20,00 GB).
- Sound:** Sound Driver: PulseAudio, Controller: ICH AC97.
- Network:** Adapter 1: Intel PRO/1000 MT Desktop (NAT).
- USB:** USB Filter Rules: 0 (0 active).
- Shared Folders:** None.
- Description:** None.

The left sidebar shows the "Ubuntu Server" VM selected, with options for "Güçü Kapat" (Power Off) and "Yeni" (New). The top menu bar includes "Dosya", "Makine", and "Yardım". The top right corner has "Ayrıntılar" (Details) and "Anlık Görüntüler" (Snapshots) buttons. The main preview window shows a black box with the text "Ubuntu Server".

VirtualBox



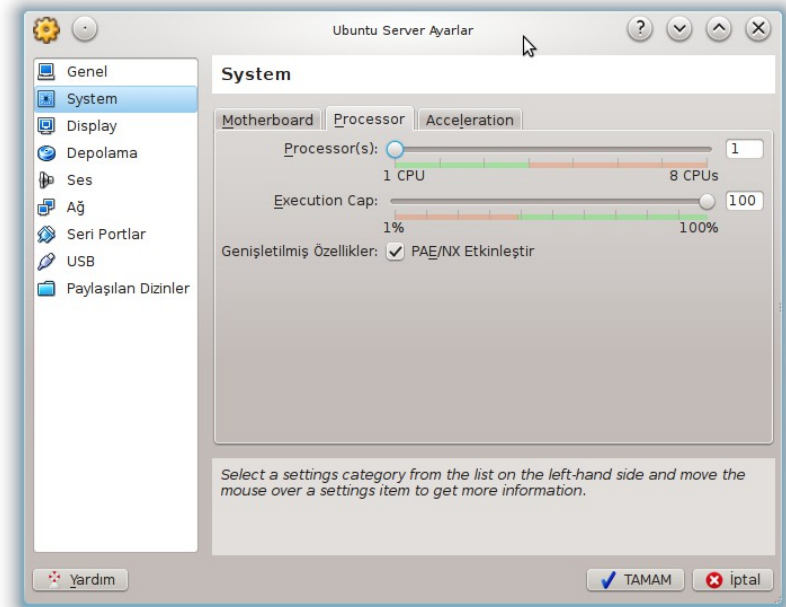
VirtualBox



- Sistem seçeneğinde sanal makinenin sanal anakartı ve bu anakart üzerindeki temel birimler olan BIOS, bellek ve işlemci konusundaki ayarlar yapılmaktadır.
- VirtualBox'ın sağladığı sanal BIOS'un en önemli özelliği sistemdeki aygıtların açılış sırasidir. Bu sıra ok tuşları ile değiştirilebilir.
- Buna ek olarak sanal anakartın çip takımı (PIIX3 veya ICH9) seçilebilir.

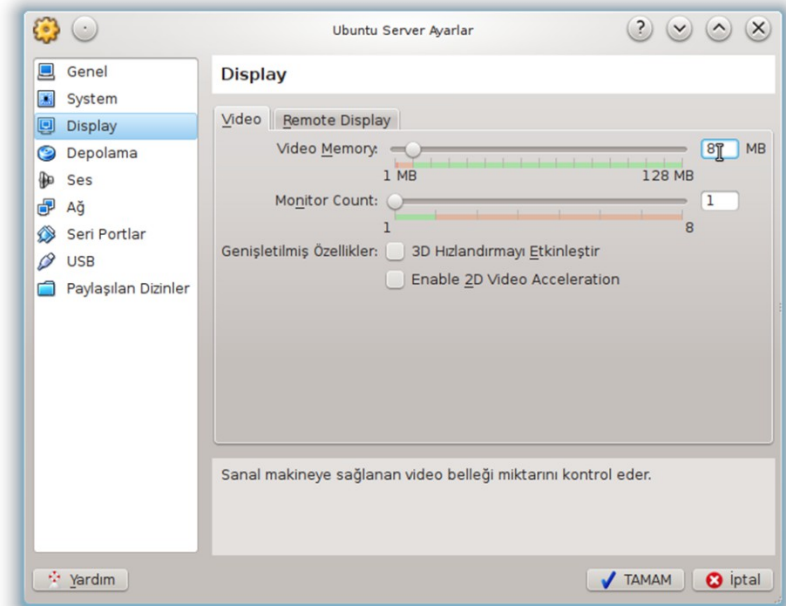
VirtualBox

- Sanal makine ayarları içerisinde performansa en çok etki eden ayarlardan birisi de atanan işlemci kapasitesidir.
- İşlemci kapasitesi dağıtılırken şunlara dikkat etmek gerekir.
 - Virtualbox için her bir işlemci çekirdeği bir adet işlemci gibi gözükecektir. Ayrıca Hyper-Threading (HT) özellikli çekirdekler 2 veya daha fazla sayıda işlemci olarak gözükebilirler. Bu nedenle bilgisayarda kolayca 8 hatta 16 işlemci varmış gibi gözükebilir.
 - Virtualbox içinden çalıştırılacak olan işletim sisteminin özellikle işlemci yoğun bir çalışma yapması düşünülmüyorsa yada grafik arayüz olarak güçlü bir sistem beklentisi yoksa, tek bir işlemci çekirdeğinin verilmesi yeterli olacaktır.
 - Çok az işlem yapacak olan sistemlerin (örneğin bir DNS sunucusu) tek bir çekirdeğin belirli bir yüzdesi ile çalışması dahi yeterlidir.



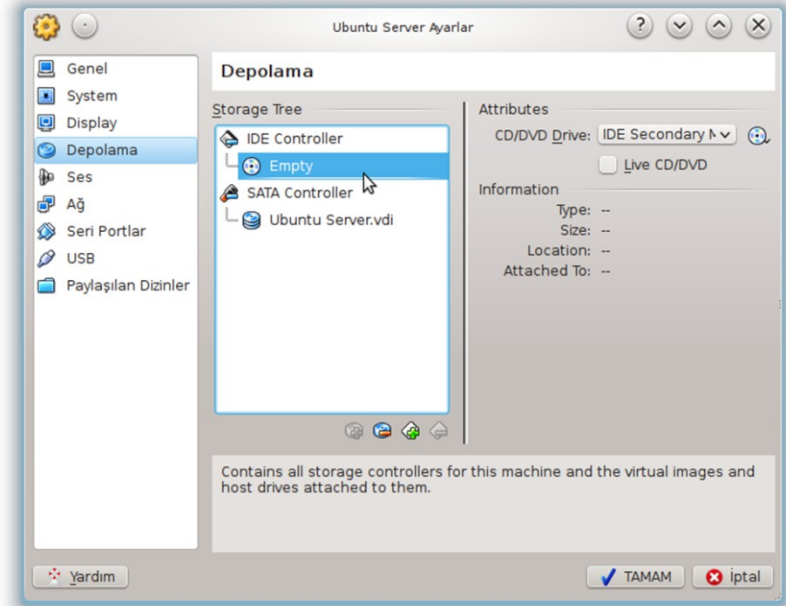
VirtualBox

- Kurulacak olan sanal makinenin ekran kartı belleği o sanal makinenin grafik kabiliyetlerini doğrudan etkileyecektir.
 - 8 MB altında belleği olan sanal makinelerin tam ekran kipinde gösterilmesi mümkün değildir. Ancak sadece çalıştırılacak ve asla tam ekran kipinde kullanılmayacak olan sanal makineler de yapılandırılabilir. Bir çok sunucu, ayrıca güvenlik duvarı gibi ağ odaklı sanal makine ayarlarında bu durum geçerlidir.
 - Tam ekran kipinde kullanılması düşünülen ancak üstün grafik arayüz özelliklerine ihtiyaç duymayan sunucular ve temel masaüstü ortamları için 8-16 MB aralığında ekran kartı belleği uygun olacaktır.



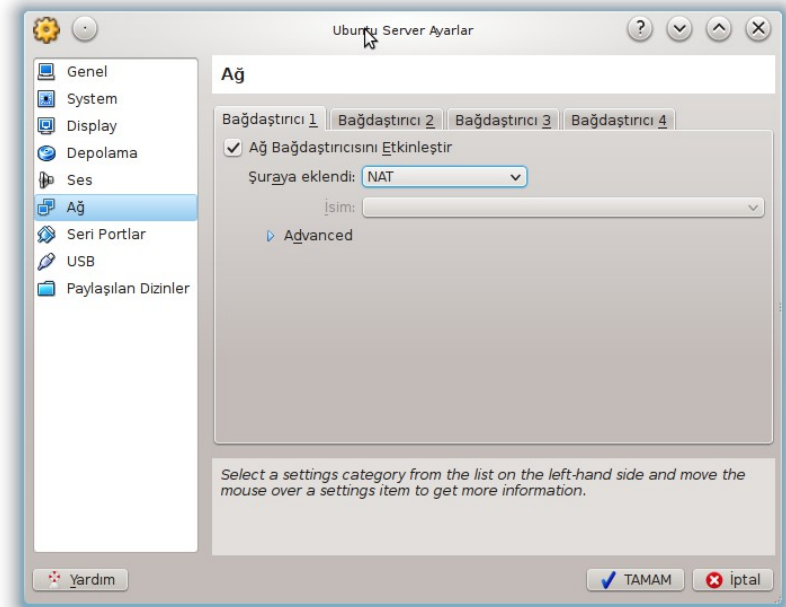
VirtualBox

- Sanal makinenin CD/DVD sürücüsüne çalıştırmadan önce ve çalışma sırasında CD/DVD imajları eklenebilir.



VirtualBox

- VirtualBox ile sanal makinelere dört adede kadar sanal ağ kartı (kablolu) tahsis edilebilir. Bu kartların ağ içinde nasıl kullanılacakları, özellikle IP adresi alma mekanizmaları da ayarlanabilir.
 - NAT seçeneğinde, sanal ağ kartı, VirtualBox tarafından işletilen sanal bir yönlendiricinin arkasında bağımsız bir IP adresi alır.
 - Köprü (Bridge) seçeneğinde sanal ağ kartı, var olan gerçek bir karta köprülenir. Bu durumda sanal ağ kartının MAC adresi kullanılarak gerçek ağda doğrudan işlem yapılabilir.
 - Dahili ağ seçeneğinde sanal ağ kartı, sadece sanal makinelerin bağlanabildiği bir iç ağa dahil olur. Bu ağdaki trafiği, ev sahibi işletim sistemi göremez.
 - Ev sahibi (Host only) seçeneğinde sanal ağ kartı, misafir ve ev sahibi işletim sistemleri arasında ağ iletişimi sağlamak için kullanılır.



6. Bölüm Güvenlik Duvarları

- Bu bölümde, güvenlik duvarları kavramlarını tanıyacak ve güvenlik duvarı kurmayı özetleyeceğiz.
- Belli başlı konular
 - Güvenlik duvarları
 - Linux IPTables alt yapısı
 - PFSense güvenlik duvarı dağıtımı

Güvenlik Duvarları

- Güvenlik Duvarı: Ağda ya da yerel makinede, ağ operasyonlarını kısıtlamaya, değiştirmeye yarayan sistemdir.
- Paketlerin şu özelliklerine göre onlarda bir işlem yapmak isteyebilirsiniz:
 - Protokol
 - Kaynak IP, kaynak port
 - Hedef IP, hedef port
 - MAC adresi
 - Ağ arabirimi
 - TCP paket bayrakları (SYN, ACK...)
 - ...

Güvenlik Duvarları

- Güvenlik Duvarı aşamaları:
 - Paket Filtreleme
 - Paketin durumu bilinmez. Bağlantı durumu gözetilmez.
 - Hızlıdır.
 - Durum Korumalı Paket Filtreleme (Stateful PF)
 - Gelen paketin daha önce açılan bir bağlantının üye paketlerinden birisi olup olmadığı kontrol edilebilir.
 - Bunun için de protokolleri bilir.
 - Devre Seviyesinde Vekillik
 - Aktarım (transport) katmanında çalışan paket filtrelerin aksine oturum katmanında çalışır.
 - Bağlantısı gerçekleşmiş istemcinin bir hizmeti alıp alamayacağına karar verir ve istemci-sunucu arasında “sağır” aracılık yapar
 - Uygulama Seviyesinde Vekillik
 - Uygulama seviyesinde çalışır.
 - İstemci-sunucu arasındaki dili bilir. İstemciden gelenleri “kulaktan kulağa” sunucuya aktarır. İçerik elinden geçtiğinden kontrol yapabilir.

Linux IPTables Alt Yapısı

- Linux'ta güvenlik duvarı geçmişi
 - ipfwadm – Çekirdek 2.0
 - ipchains – Çekirdek 2.2
 - iptables – Çekirdek 2.4 ve üzeri
- Linux'da güvenlik duvarı alt yapısı çekirdek modülleri üzerinden çalışmaktadır.

Linux IPTables Alt Yapısı

- Netfilter zincirlerden oluşur. Ön tanımlı zincirler:
 - PREROUTING
 - INPUT
 - FORWARD
 - OUTPUT
 - POSTROUTING
- Bu zincirler ön tanımlı tablolarda kullanılırlar:
 - nat
 - filter
 - mangle

Linux IPTables Alt Yapısı

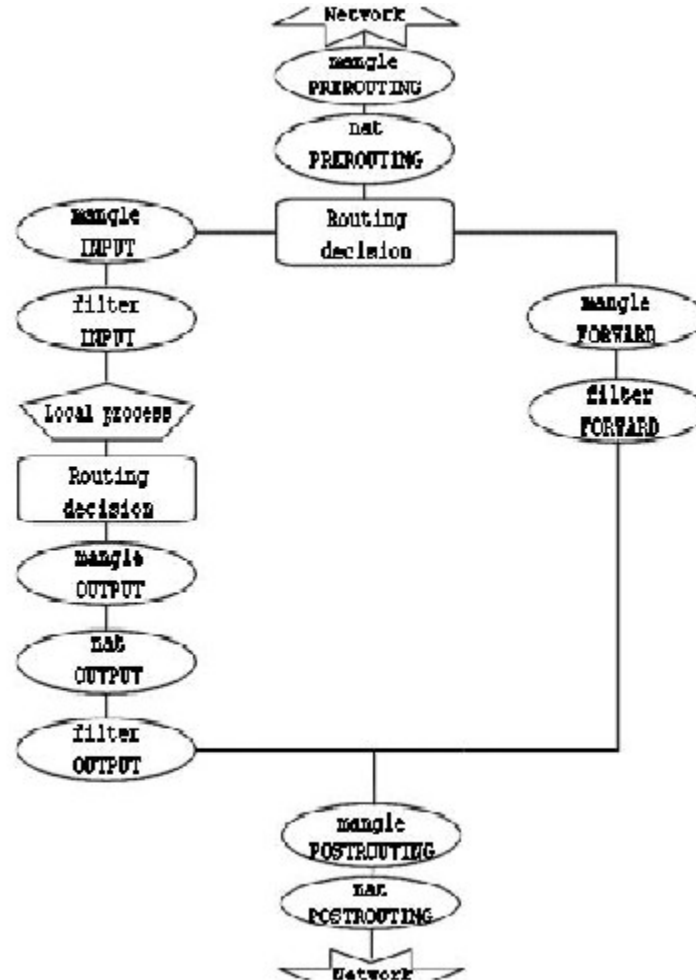
1			Ağ arabirimine gelir
2	mangle	PREROUTING	Paket değiştirme
3	nat	PREROUTING	DNAT, REDIRECT
4			Yönlendirme
5	mangle	INPUT	Paket değiştirme
6	filter	INPUT	Filtreleme
7			Verinin hedef sahibi yazılım

1			Ağ arabirimine gelir
2			Yönlendirme
3	mangle	OUTPUT	Paket değiştirme
4	nat	OUTPUT	SNAT, MASQUERADE
5	filter	OUTPUT	Filtreleme
6	mangle	POSTROUTING	Paket değiştirme
7	nat	POSTROUTING	SNAT, MASQUERADE
8			İlgili ağ arabiriminden çıkış

Linux IPTables Alt Yapısı

1			Ağ arabirimine gelir
3	mangle	PREROUTING	Paket değiştirme
4	nat	PREROUTING	DNAT, REDIRECT
5			Yönlendirme
6	mangle	FORWARD	Paket değiştirme
7	filter	FORWARD	Filtreleme
8	mangle	POSTROUTING	Paket değiştirme
9	nat	POSTROUTING	SNAT, MASQUERADE
10			İlgili ağ arabiriminden çıkış

Linux IPTables Alt Yapısı



Linux IPTables Alt Yapısı

- Tablolar
- Mangle:
 - Paketlerin TTL, TOS, MARK bölümlerinde değişiklik yapılabilir.
 - TTL: Time to Live
 - TOS: Type of Service: Paketin içerdiği veri cinsi vb. (çoklu ortam, normal veri vs.)
- MARK: Bu kullanılarak yönlendirici cihazlara yönlendirme kararları ile ilgili ipucu verilebilir. Bant Genişliği Sınırlaması ve Sınıf Bazlı Kuyruklar da bu işaretler ile oluşturulur.

Linux IPTables Alt Yapısı

- NAT: Ağ Adres Dönüşümü (Network Address Translation)
- Paketlerin Hedef-kaynak IP-port deperlerini değiştirmek için kullanılır
- DNAT: Paketin hedef adresini değiştirir.
- `$ iptables -t nat -A PREROUTING -p tcp -d 144.122.149.1 --dport 80 -j DNAT --to-destination 192.168.0.6 --to-destination`
- `192.168.0.6-192.168.0.10`: Hedefi 5 sunuculuk bir havuza yönlendirdik.
- `--to-destination 192.168.0.6:81`: Hedef portunu değiştirdik.

Linux IPTables Alt Yapısı

- NAT: Ağ Adres Dönüşümü (Network Address Translation)
- İçerideki sanal IPsi olan bir sunucuya gelen paketleri sunucuya ulaştırmak için kullanılabilir.
- REDIRECT:Özelleşmiş DNAT tanımı
 - `$ iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080`
- Yerel makinede portlar arası yönlendirme

Linux IPTables Alt Yapısı

- SNAT: Kaynak adresini değiştirmek için kullanılır.
- `$ iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to-source 144.122.149.1 --to-source 144.122.149.1-144.122.149.5:`
- 5 IP adresine dengelendi.
- İç ağdaki sanal IP'li makineleri internete çıkarırken kullanılabilir.

Linux IPTables Alt Yapısı

- MASQUERADE: --to-source gerektirmeyen SNAT.
- NAT işlemlerini yapan makinenin DHCP ya da dial-up ile dinamik IP'si olması durumunda kullanılır.
- `$ iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000`

Linux IPTables Alt Yapısı

- Filter: Paket filteleme işlemleri için kullanılır. Paketleri geçirip geçirmemek konusunda esas kararların alındığı tablodur. Kararlara göre çeşitli aksiyonlar seçilir.
- Durum Makinesi(Bağlantı takip makinesi): Bağlantıların durumlarına göre paketlere durum atayarak tüm akışı takip eden çekirdek modülüdür.

Linux IPTables Alt Yapısı

- Daha akıllı filtreleme işlemleri için durum makinesi önemlidir. Bağlantılar durumlarına göre şu sınıflara ayrılırlar:
 - NEW: Paket duvar için yeni bir paket ise durum “NEW” olarak nitelendirilir. (ör: TCP bağlantısı aşamasında gelen bir SYN paketi)
 - ESTABLISHED: İki makine arasında konuşma başlamış ise paketler “ESTABLISHED” olarak değerlendirilir. Örneğin SYN’ye cevap olarak gönderilen ACK’dan sonra durum bu olacaktır.
 - RELATED: Kurulmuş bir “ESTABLISHED” durumundaki bağlantı ile ilgili bir bağlantıya ek olarak bir bağlantı açılmış ise bu durum gerçekleşecektir. Örneğin ilk kurulan FTP bağlantılarını ardından bağlantının farklı portalara taşınması olayında yeni bağlantı “RELATED” durumda olacaktır.
 - INVALID: Bir şekilde hiçbir duruma girmemiş paketlere verilir. Sistem hafızası yetmediğinde ya da gelen ICMP mesajlarına bir anlama verilemediğinde oluşabilir.

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **Komutlar:**
 - **-A, --append:** Belirtilen zincirin sonuna bir kural ekler.
\$ iptables -A INPUT
 - **-I, --insert:** Belirtilen zincirin başına bir kural ekler.
\$ iptables -I INPUT
 - **-D, --delete:** Belirtilen zincirden bir kural siler.
\$ iptables -D INPUT -s 192.168.0.4 -j DROP
\$ iptables -D INPUT 1
 - **-R, --replace:** Belirtilen zincirden bir kuralu değiştirir.
\$ iptables -R INPUT 1 -s 192.168.0.4 -j DROP
 - **-L, --list:** Belirtilen zincirdeki kuralları listeler.
\$ iptables -L INPUT
 - **-F, --flush:** Belirtilen zincirdeki tüm kuralları siler.
\$ iptables -F INPUT

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **Seçenekler:**
 - -v: Ayrıntılı çıktı
 - -n: IP'leri ve portları sayısal listele
 - **Eşleme (match) kuralları:**
 - -p, --protocol
 - -s, --source
 - -d, --destination
 - -i, --in-interface
 - -o, --out-interface
 - -f, --fragment

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **Dolaylı Eşleme (match) kuralları:**
 - **-TCP Eşleme (match) kuralları:**
 - **--sport, --source-port**
 - **--dport, --destination-port**
 - **--tcpflags**
 - **--tcp-option**
 - **-UDP Eşleme (match) kuralları:**
 - **--sport, --source-port**
 - **--dport, --destination-port**
 - **-ICMP Eşleme (match) kuralları:**
 - **--icmp-type**

Linux IPTables Alt Yapısı

- **\$ iptables**
 - **\$ iptables [-t *table*] command [match] [target/jump]**
 - Doğrudan Eşleme (match) kuralları:
 - Limit Eşleme (match) kuralı:
 - -m limit --limit 3/second
 - -m limit --limit-burst 5
 - MAC:
 - -m mac --mac-source 00:06:cd:fe:00:00
 - MARK:
 - -t mangle -m mark --mark 1
- \$ iptables -t mangle -A INPUT -m mark --mark 1**

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **Multiport eşlemesi:**
 - **-p tcp -m multiport --source-port 110,143**
 - **-p tcp -m multiport --destination-port 53,80**
 - **-p tcp -m multiport --port 80,110 : Aynı portlar arasında veri transferi var ise.**
 - **Sahip Eşlemesi:**
 - **-m owner --uid-owner 500**
 - **-m owner --gid-owner 500**
 - **-m owner --pid-owner 5453**
 - **-m owner --sid-owner 100**

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **Durum eşlemesi:**
 - **-m state --state RELATED,ESTABLISHED**
 - **TOS eşlemesi:**
 - **-p tcp -m tos --tos 0x16**
 - **\$ iptables -m tos -h : tipler**
 - **TTL eşlemesi:**
 - **--ttl-eq ttl**
 - **--ttl-gt ttl**
 - **--ttl-lt ttl**
 - **-m ttl --ttl 60**

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - Hedefler
 - **-j JUMP**
\$ iptables -A INPUT -p tcp -j yenizincir1
 - **-j TARGET**
 - **ACCEPT:** Paket bu tablodaki bir zincirin bir kuralında kabul edilir. Bu zincirin geri kalanı ve bu tablonun geri kalanı gezilmez. Ancak paketin hala diğer tablolardaki(nat,filter,mangle) zincirlerde durdurulma olasılığı vardır.
 - **DROP:** Sessiz sedasız paket durdurulur. Paket diğer zincir ve tabloları dolaşmaya devam etmez.
 - **REJECT:** Paket reddedilir ve reddedildiği bildirilir. (ICMP tipleri ile)
\$ iptables -I INPUT -p tcp -destination-port 22 -j REJECT -reject-with tcp-reset

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **-j TARGET**
 - **RETURN:** Bu zincirde işi bitmiştir. Bir üst zincir ya da diğer tablodan hiçbirşey olmamış gibi devam eder.
 - **TOS:**
 - \$ iptables -t mangle -A PREROUTING -p TCP --dport 22 -j TOS --set-tos 0x10**
 - **TTL:**
 - \$ iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-set 64**
 - \$ iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-dec 1**
 - \$ iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-inc 1**

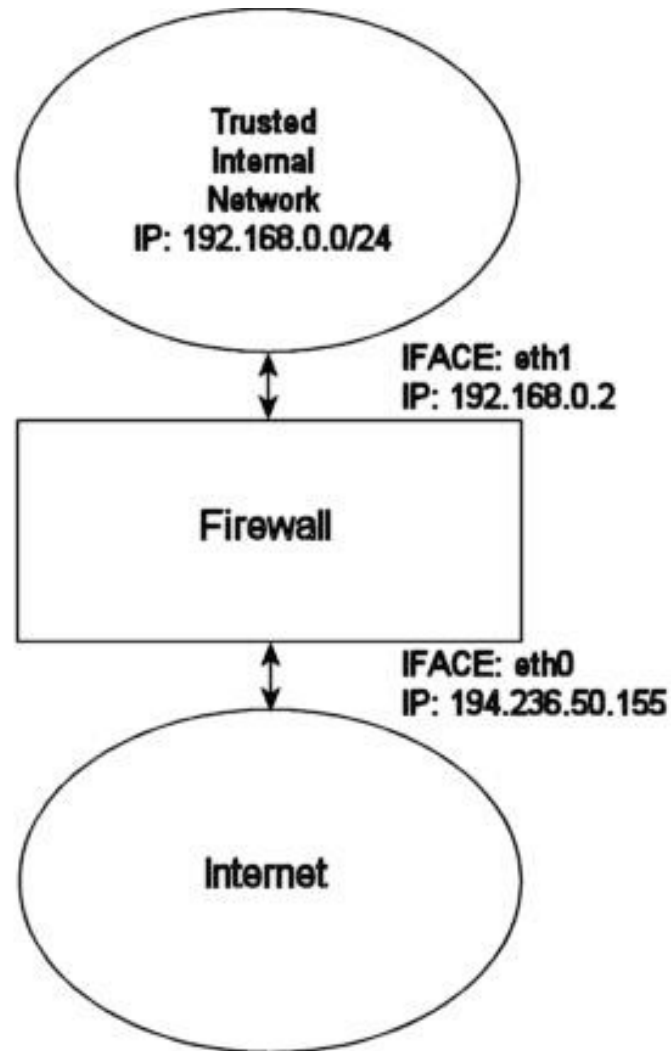
Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **-j TARGET**
 - **LOG:**
 - \$ iptables -A FORWARD -p tcp -j LOG --log-level debug**
 - \$ iptables -A INPUT -p tcp -j LOG --log-prefix "INPUT packets"**
 - \$ iptables -A INPUT -p tcp -j LOG --log-tcp-sequence**
 - \$ iptables -A FORWARD -p tcp -j LOG --log-tcp-options**
 - \$ iptables -A FORWARD -p tcp -j LOG --log-ip-options**

Linux IPTables Alt Yapısı

- **\$ iptables**
- **\$ iptables [-t *table*] command [match] [target/jump]**
 - **-j TARGET**
 - **DNAT:**
\$ iptables -t nat -A PREROUTING -p tcp -d 15.45.23.67 --dport 80 -j DNAT --to-destination 192.168.1.1-192.168.1.10
 - **SNAT:**
\$ iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to-source 194.236.50.155-194.236.50.160:1024-32000
 - **MASQUERADE:**
\$ iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000 0
 - **REDIRECT:**
\$ iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080

Linux IPTables Alt Yapısı



PFsense Güvenlik Duvarı Dağıtımı

- PFSense, BSD UNIX temelli yaygın bir güvenlik duvarı dağıtımıdır. Uygulamada sık kullanıldığı için bu eğitim kapsamında kısaca bahsedilecektir.
- PFSense'in avantajları şunlardır:
 - Uzun süredir kullanımda olması ve test edilmiş yapısı
 - Çok sayıda modülü ile farklı amaçlara odaklanabilmesi
 - Son derece düşük işlemci ve bellek kullanımı
 - Sanallaştırma yolu ile sanal makineler için kullanılabilmesi
 - Kolay kullanılır web arayüzleri
- Öte yandan, PFSense veya benzeri bir güvenlik duvarı dağıtımını iyi kullanabilmek için temel güvenlik duvarı kabiliyetlerini iyi tanımak gereklidir. Bu nedenle IPTables ile çalışmış olmak PFSense'i de daha iyi kullanma sonucunu doğruacaktır.

7. Bölüm

Özgür Yazılıma Katkı Sağlamak

- Bu bölümde, özgür yazılımlara nasıl katkı sunulacağına dair bazı öneriler yapılmaktadır.

Katkı Sağlamak

- Özgür yazılım bir süreçtir ve kullanıcıların katkıya dönülmesi sayesinde canlı ve hareketli bir yapı kazanır.
- Bu nedenle bu eğitime katılan kişilerin de kendi olanakları ölçüsünde Linux ve özgür yazılımlara katkı sağlamayı değerlendirmesi beklenmektedir.
- Bu katkılar bir çok şekilde oluşabilir.
 - Kullanıcı desteği
 - Yaygınlaştırma
 - Belgeleme
 - Çeviriler
 - Test
 - Geliştirme

Katkı Sağlamak

- Kullanıcı desteği
 - Var olan kullanıcıların sorunlarına çözüm getirmek yolu ile Linux ve özgür yazılım kullananlara destek olmak son derece önemlidir.
 - Burada dikkat edilecek bir konu tamamen linux ve özgür yazılım alt yapısına geçemeyen kullanıcıların da Libre Office, Thunderbird, vb özgür yazılımlardaki ihtiyaçlarının giderilmesidir.
- Yaygınlaştırma
 - Yaygınlaştırma bir çok biçimde olabilir. Bireysel kullanımdan büyük kurumsal göçlere kadar değişik ölçeklerde veya tek bir uygulamadan tüm uygulama alanlarına kadar değişik kapsamlarda olabilir.
 - Yaygınlaştırma sırasında dikkat edilmesi gereken şey, ilk aşamada bir ihtiyaç analizi yapılmasıdır. İhtiyaç analizi sonucunda emin olunarak başlatılan yaygınlaştırma çalışmaları daha başarılı olmaktadır.

Katkı Sağlamak

- Belgeleme
 - Belgelemeden kast edilen bir çok şey olabilir. Uygulamaların içinde gelen yardım dosyalarından başlayarak İnternet üzerinde yer alan çeşitli ortamlardaki belgelere kadar bir çok yerde işe yarar belgeler bulmak mümkündür.
 - Belgelemede odaklı çalışan gruplara katılmak kadar kendi ihtiyaçlarınızı nasıl giderdiğinizi anlatan kişisel tarihçe türü belgeler oluşturmak da yararlıdır.
 - Odaklı çalışan gruplar bazen tek bir uygulamaya bazen de bir alandaki (örneğin bir meslek grubu) uygulamalara genel olarak odaklanmaktadır.
 - Belirli bir dağıtım için oluşturulan belgenin içeriğinin farklı bir dağıtımda birebir uygulanması bazen mümkün olmaz. Bu durumda belgenin içinde bu konudan bahsetmek ve diğer dağıtımlarda nelerin değişebileceğini ifade etmek doğru olacaktır.

Katkı Sağlamak

- Çeviriler
 - Özgür yazılım sınırları aşan bir yapıya sahiptir. Ancak dil engeli yine de karşılaşılan önemli bir sorundur. Bu nedenle çeviriler önemlidir.
 - Çevirilerde genellikle belirli bir dile yada belirli bir uygulamaya odaklanmış gruplara katılmak iyi bir başlangıç noktasıdır. Bu durumda uygulamanın menüleri, yardım sayfaları, kurulum belgeleri gibi standart bazı parçaları çevirilir.
 - Buna ek olarak yaygın atıfta bulunulan yabancı belgeleri yazarından izin alarak Türkçe'ye çevirmek de iyi bir faaliyet türüdür.
 - Çeviri gruplarının bazı avantajları bulunur. Özellikle şu iki konuda çeviri grupları daha verimli çalışır.
 - Çeviride kalite güvence çok önemlidir. Bitmiş bir çevirinin incelenmesi ve hatalı çevirilerin düzeltilmesi için çevirmenlerle iletişime geçilmesi de önemli bir katkıdır.
 - Bir diğer önemli konu da çevirilerin güncelliğidir. Bu konuda da çevirilen bir belgenin veya yazılım varlığının yeni sürümlerinin çıkıp çıkmadığının takip edilmesi gerekir.

Katkı Sağlamak

- Test
 - Özgür yazılımların test edilmesi kullanıcıların da katıldığı bir süreçte yürütülür.
 - Bir yazılımı kullanırken yaşanan sorunların raporlanması ve bu sorunların hangi şartlar altında yaşandığının tespit edilmesi için aynı sorunun yeniden yaratılması için denemeler yapılması test sürecinin giriş aşamasıdır denebilir. Bu aşamada dikkatli çalışarak detaylı bir belge oluşturup yazılımın geliştiricilerine ulaştırılması gerekir.
 - Bunun yanı sıra, yeni bir özelliğin doğru çalışıp çalışmadığının anlaşılması yada daha önceden belgelenmiş bir sorunun giderilip giderilmediğinin anlaşılması için testler yapılması da mümkündür.
 - Testler için bağımsız gruplar oluşturulabileceği gibi doğrudan yazılımı geliştiren ekiple eşgüdüm içinde çalışmak da mümkündür.
- Geliştirme

Katkı Sağlamak

• Geliştirme

- Özgür yazılıma sağlanabilecek olan en büyük katkı, yazılım geliştirmektir. Ancak bu katkı aynı zamanda en büyük emeği içerdiği için daha az kişinin yapabileceği bir şeydir.
- Bu tür katkıyı sağlayacak olan kişi kendisinin ayırabileceği zamanı ve özellikle sürekli olarak katkı sağlayıp sağlayamayacağını değerlendirmelidir.
 - Sürekli olarak katkı sağlayamayacak olan kişilerin daha çok yazılımların detaylı testleri, yamaların uygulanması ve kontrolü türü kısa süreli uğraşlarla destek olması doğru olacaktır.
 - Sürekli olarak katkı sağlayabilecek olanlar ise öncelikle kendi kullandıkları uygulamalardan başlayarak uygulamaları geliştiren ekipler ile bağlantıya geçerek bir «iş ataması» talep etmelidir.
- Özgür yazılım projelerinde «o projeye dönük» liyakat çok önemlidir. Bu nedenle yeni geliştiricilere öncelikle küçük ve düşük riskli işler verilmesi normaldir. Kişinin projeye olan katkısı arttıkça ve projeye ayırabildiği zamanın sürekli olduğu görüldükçe yapılan işlerin ölçeği ve etkisi de artacaktır.

Katkı Sağlamak

- Geliştirme

- Elbette her zaman için yeni bir özgür yazılım için proje geliştirilmesi de gündemde olabilir.
- Ancak bir özgür yazılım projesinin yürütülmesi karmaşık bir iştir. Bu nedenle öncelikle başka bir projede bu konularda deneyim kazanılması, hayata geçirilmesi düşünülen projenin selameti için önerilmektedir.

8. Bölüm

Daha Sonrası

- Bu bölümde bu eğitimde elde edilen bilgi ve becerilerin ne şekilde ilerletilebileceđi konusunda kısa bazı öneriler yapılacaktır.

Daha Sonrası

- Her türlü becerinin «korunması» için o beceri üzerinde pratik yapılması gereklidir. Bu eğitim de farklı değildir.
- Bu nedenle katılımcıların öğrendikleri uygulamaları mümkün olduğunca kullanmaları ve özellikle işyerlerindeki yerel ağlara dair ihtiyaçları Linux temelli olarak karşılamak için çaba göstermeleri önemlidir.
- Buna ek olarak belirli ihtiyaçlara odaklı olarak bazı konularda ek çalışmaların yapılması önerilebilir.

Daha Sonrası

- Linux sistem yönetimine dair pratik becerilerin artırılması için ise aşağıdaki uygulamalar yararlı olacaktır.
 - BASH betik programlama becerilerinin artırılması için hemen her konuda betikler yazılması gereklidir.
 - BASH dışında betik yazmak ve hazır yazılmış betikleri kullanmaya alışmak için Python ve Perl dillerinden en az birisini öğrenmek yararlı olacaktır.
 - Çok sayıda bilgisayarın uzaktan yönetilmesi konusunda deneyim kazanmak için «puppet» adındaki aracın kullanımı hem sistem yönetim sürecini hem de kullanılan araçları daha iyi tanımayı sağlayacaktır.