Kim Korkar UNIX'ten?

Can Uğur Ayfer

Aralık 1995

İçindekiler

ÖNSÖZ

Bugüne kadar 1000'e yakın sayıda farklı bilgisayar modeli üretildiği sanılıyor. Bu sayının içinde, binlerce değişik firma tarafından üretilen PC'ler tek bir model olarak yer almaktadır.

Bu kadar geniş donanım yelpazesi içinde yüzlerce değişik **işletim sistemi** geldi geçti. Adını bilgisayar tarihine altın harflerle yazdırmayı başarabilenlerden biri de UNIX oldu..

Oldu ama pek fazla da sempati toplayamadı. UNIX'le bir şekilde ilgilenen ya da ilgilenmek zorunda kalan pek çok kişiden duyduklarım genellikle UNIX'in sevimsiz, kullanması zor, kaprisli bir işletim sistemi olduğu doğrultusunda oldu. İtiraf edeyim ki, UNIX'le ilk tanıştığım 1983 yılında benim de görüşüm bu yöndeydi.

Uzun yıllar ticari uygulamalarda, yalnızca "çok kullanıcılık" uğruna insanlar UNIX'e katlandılar. Derken, UNIX altında grafik ekran kullanımını sağlayan X-Windows ortaya çıktı; hemen ardından bilgisayar ağlarının ve doğal olarak Internet'in yıldızı parladı. İşte o zaman kullanıcılar ve programcılar UNIX'i bir daha değerlendirme gereksinimi duydular.

İçinde bulunduğumuz yıllarda UNIX çok önemli bir işletim sistemi! UNIX'le konuşamayan, TCP/IP desteği olmayan bilgisayar ağı yazılımları satamıyor; bir çok kişisel bilgisayar yazılımının UNIX uyarlamaları var. Kısacası UNIX'in gelişmesi ve yaygınlaşması hızlanmış durumda. Önümüzdeki bir kaç yıl içinde, mesleği bilgisayar kullanımı gerektiren herkesin, ucundan da olsa UNIX'e bulaşmadan çalışmasının olanaksız olacağı görüşü oldukça yaygın.

Bu durumda, bilgisayar dünyasına kişisel bilgisayarlarla adım atmış kullanıcı kitlesine UNIX'i tanıtmak ve hazırlıklı olmalarına yardımcı olabilmek amacıyla bu kitabı yazmaya başladım. Önce, DTK şirketince üretilmekte olan SPARC serisi iş istasyonları için notlar halinde ortaya çıkan bir döküman zamanla elinizdeki bu kitaba dönüştü.

Bu kitap UNIX hakkındaki her şeyi anlatmıyor; zaten sonlu sayıda sayfa kullanılarak UNIX hakkındaki her şeyi anlatmak da pek olası değil! Tek amacım, UNIX hakkında ön yargısı veya kötü deneyimleri olanlara UNIX'in kötü bir işletim sistemi olmadığını; aslında bir sanat eseri olduğunu; iyi kullanmayı bilen birisinin elinde neler yapabileceğini anlatmak. Ne demişler; at binenin ...

Kitapta anlatılanları izleyebilmek için, en azından MS-DOS işletim sistemi konusunda deneyimli olmanız gerekiyor. Bu kitap, bilgisayarlar hakkında genel bilgi arayışı içinde olan okuyucular için hiç de uygun değil.

Bütün bilgisayar kitaplarında olduğu gibi, bu kitabı da okurken, anlatılan komutları ve örnekleri kendi bilgisayarınızda denemelisiniz. Ancak, UNIX dünyasında bu henüz pek kolay değil. Nedeniyse, henüz evlerdeki bilgisayarlara UNIX'in girmemiş olması. Kitabın ekindeki disketin içinde, bu kitapta adı geçen UNIX komutlarının bir kısmının PC'lerde, MS-DOS altında çalışabilen modellerini bulacaksınız. "Modellerini" diyorum; çünkü bu MS-DOS programları, tam tamına UNIX karşılıklarının eşdeğeri değil. Ama, gene de, okuyucunun evindeki ya da bürosundaki kişisel bilgisayarda denemeler yapması için yeterli olacağı inancındayım.

Kitabın düzenlenişini biraz garip bulabilirsiniz. Bilgisayarın açılışını ve kapanışını kitabın ortasından sonra bir yerlerde anlattım. İlk bakışta, bu konuların en başta anlatılması gerekiyor gibi düşünebilirsiniz ama tipik UNIX kullanıcıları bilgisayarı hiç açıp kapatmazlar ki... Gene de kitabın düzeninin kusursuz olduğunu savunmuyorum. Konuları bana doğal geldiği şekilde sundum; ancak kitap bitince bir de baktım; bir UNIX referans kitabından çok, UNIX hakkında bir macera kitabına benzemiş. O nedenle korkarım başından sonuna kadar okumanız gerekecek. Aslında, UNIX ve komutları hakkında referans kitaplarını her yerde bulabilirsiniz; hatta ekranınızda bile...

Bazı konuların bir kaç yerde tekrarlandığını göreceksiniz. Bu tekrarları özellikle yaptım. UNIX'de bir komut ya da kavramın öneminin ilk karşılaşıldığında iyi anlaşılamayacağını biliyorum; kendim yaşadım. O nedenle, okuyucunun da başına aynı şeyin geleceğini düşünerek, bazı konuları, öneminin vurgulanabileceği bir yere gelince tekrarlamaktan kaçınmadım.
Bu kitabı yazarken verdiği destek ve katkıları için eşim Reyyan Ayfer'e; müsveddeleri büyük bir dikkatle okuyan ve çok değerli katkılarda bulunan arkadaşım Lale Morgül'e ve bir çok yanlışımı bularak düzelten oğlum Ömer Ayfer'e çok çok teşekkür ederim.

Can Uğur Ayfer
Kasım 1995, Ankara

İŞLETİM SİSTEMLERİNİN KRALİÇESİ
UNIX

UNIX dünyasına hoşgeldiniz.

Nerelerde kaldınız? Hiç gelmeyeceksiniz sanmıştık...

Dünyada hiç bir işletim sistemi, UNIX kadar uzun ve sürekli gelişerek gündemde kalmayı başaramamıştır. IBM PC'ler için geliştirilmiş olan MS-DOS kadar yaygın olmamakla birlikte, dolaylı yoldan da olsa, UNIX işletim sisteminin hizmet vermekte olduğu kullanıcı sayısının, MS-DOS kullancılarının sayısına yakın olduğu sanılmaktadır.

UNIX işletim sistemi genellikle güçlü bilgisayarlarda kullanılmaktadır. UNIX felsefesinin temelinde, bir bilgisayarın birden fazla kullanıcı arasında paylaşılması; ya da bir kullanıcının aynı anda birden fazla iş yapmasına olanak sağlamak yatmaktadır. Bu nedenle, UNIX altında kullanılacak bilgisayarın, kaynaklarının birden fazla iş arasında paylaşılması durumunda performansını kabul edilebilir düzeyde tutabilecek güçte olması gerekmektedir. Bilgisayar teknolojisindeki gelişmeler, donanımları hızla güçlendirmekte ve ucuzlatmaktadır; bunun doğal sonucu olarak da, UNIX işletim sistemi denetiminde kullanılan bilgisayarların sayısı hızla artmaktadır.

Çok İş; Çok Kazanç...

UNIX İşletim Sistemi, bilgisayar bilimcilerinin **'çok kullanıcılı'** *(multi-user)* ve **'çok işli'** *(multi-tasking)* adını verdikleri çalışma koşullarını sağlar. Bir başka deyişle; UNIX altında çalışan bir bilgisayarı, birden fazla kullanıcı birbirlerinden bağımsız olarak ve aynı anda kullanabilirler. Bu birlikte kullanım sırasında, bilgisayarın kaynaklarını (merkezi işlem birimini, ana belleğini (RAM), disk-teyp gibi yan bellek birimlerini, yazıcılarını) paylaşırlar. UNIX, kaynakların kullanımını, paylaşımdan kaynaklanan performans düşmelerini en aza indirgeyecek şekilde düzenlemeye çalışır. Bu tür paylaşımlar, donanıma

yapılan yatırımı bir miktar azaltacağı için bir kazanç unsurudur. Yan bellek paylaşımıysa kayıtlı veri ve programları da paylaşmak demektir ki; bu da değeri oldukça yüksek başka bir kazançtır.

Bir kullanıcının aynı anda birden fazla iş yapabilmesi de bir başka kolaylıktır. Bilgisayarınızda uzun bir iş başlattığınızı varsayalım; ancak bu iş, her bir kaç dakikada bir sizin klavyeden müdahale etmenizi gerektirsin. Bu durumda, bu uzun işi başlatıp yemeğe gidemezsiniz. Tek iş düzeninde kullanım için tasarlanmış bir işletim sistemi kullanıyorsanız (MS-DOS gibi), söz konusu programın yaptığı iş tamamlanıncaya kadar bilgisayarın karşısında oturmak zorundasınız. Eğer bu işi UNIX altında çalışan bir bilgisayarda yapıyorsanız, uzun programınız bir yandan işinizi yaparken, siz öte yandan (gene aynı ekran ve klavyeyi kullanarak) bir başka iş yapabilirsiniz. Eğer başka işiniz yoksa, ikinci iş olarak bir oyun başlatıp, bekleme sürenizin biraz daha zevkli geçmesini sağlayabilirsiniz. Örneğin, bu tür beklemelerde, **Internet** üzerinde bir gezintiye çıkabilirsiniz (internet : neredeyse tüm dünyaya yayılmış olan bilgisayar ağı, *Information Super Highway).*

UNIX Her Yerde Aynı UNIX...

İlk kez 1970 yılında ortaya çıkan UNIX işletim sistemi, ticari bir amaçla tasarlanmamıştı; bu yüzden, bu yeni işletim sistemine ilgi duyan tüm bilgisayar üreticilerine ve bilgisayarını UNIX desteği ile kullanmak isteyen herkese çok küçük bir ücret karşılığında dağıtıldı. Bu sayede, UNIX kısa sürede gelişti ve yayıldı. Bu gelişmelere katkıda bulunan bilgisayarcılar, UNIX'in ilk günlerinde ortaya atılan standartları gelenekleştirerek korudular. Böylece bir **UNIX Kültürü** ve sağlam bir **UNIX Geleneği** oluştu. Kullanıcılar açısından bunun anlamı oldukça basit : UNIX İşletim sistemini bir kez öğrendiniz mi, UNIX'le çalışan herhangi bir bilgisayarı kolaylıkla kullanabileceğiniz gibi; alışık olduğunuz komut ve kavramların yüzde 99'u farklı bilgisayarlarda bile aynen geçerli kalacaktır.

Çok İyi Tasarımlanmış Bir İşletim Sistemi

UNIX İşletim Sistemi'nin 25 yıllık bir geçmişi var. Bu süre bilgisayar endüstrisi için çok ama çok uzun. Son 25 yıl içinde bilgisayarlar çok değişti, gelişti, hızlandı, küçüldü; 25 yıl önceki donanım tasarımları çoktan unutuldu ama UNIX İşletim sistemi, ilk yıllarında sahip olduğu özellikler ve yeteneklerle dimdik ayakta duruyor. 25 yaşındaki yaşlı UNIX, (belkide sadece 'olgun' demek daha doğru) günümüz bilgisayar-larına çok kolay uyum sağladı ve bundan sonraki gelişmelere de rahatça ayak uydurabilecek gibi görünüyor.

Kraliçe, Çünkü Herkes Saygı Duyuyor

Bilgisayar dünyası, yaklaşık 50 yıllık tarihinin son 30-35 yılında, **İşletim Sistemleri**'ne bir çok örnek gördü geçirdi. Bunlardan bazıları çok başarılı oldu, bazıları özel uygulamalara hizmet etti ve ömrünü tamamladı, bazıları piyasaya çıkamadan yok oldu, unutuldu gitti. İşletim sistemleri genellikle donanım üreticileri tarafından, ürettikleri bilgisayar modelleri için özel olarak geliştirildiler. Söz konusu donanım modelleri ortadan kalktıkça, bu bilgisayarların işletim sistemleri de sahneden ayrıldılar.

UNIX için böyle olmadı; çünkü belirli bir marka veya model donanım için tasarımlanmamıştı. Donanım modelleri geliştikçe, UNIX bu yeni platformlara uyarlandı ve eski deneyim, yazılım birikimleri zarar görmeden yeni bilgisayar nesillerine taşındı. Bu uyumluluğun yararını gören bilgisayar üreticilerinin neredeyse tamamı, işletim sistemi repertuarlarına UNIX'i eklemek zorunluluğunu hissettiler. Hatta bir çok bilgisayar üreticisi dev firma, kendi UNIX türevlerini geliştirdiler. AIX (IBM), ULTRIX (DEC), HPUX (HP), SINIX (SIEMENS) gibi...

Biraz da Tarih...

1960'lı yıllarda kullanılan bilgisayarlar, ancak 'Sıralı İş Düzeni'nde çalışabilmekteydi. *(Batch Processing).* Bir diğer deyişle, kullanıcılar ve programcılar, bilgisayarda yapmak istedikleri işle ilgili komut ve/veya programları bilgisayarın operatörüne teslim ederler ve sıranın kendi işlerinin yapılmasına gelmesini beklerlerdi. Bu sıra artık onbeş dakikada mı yoksa üç günde mi gelir, bilinmezdi.

Bu yıllarda, üç önemli kuruluş (AT&T, MIT Üniversitesi ve General Electric) bir arada yürüttükleri bir projeyle ilk **'Zaman Paylaşımlı İşletim Sistemi'** üzerinde çalışmaya başladılar. Proje, bir bilgisayarın bir anda birden fazla kullanıcıya hizmet etmesini sağlayan; kullanıcıların bilgisayar programında olup bitenleri izleyebileceği ve programlarla etkileşimli *(interactive)* olarak çalışabilecekleri bir ortam yaratmaya yönelikti. Çalışmalar sonunda **MULTICS** işletim sistemi ortaya çıktı *(**MULT**iplexed **I**nformation and **C**omputing **S**ystem).* Her şey akademik olarak çok iyiydi; fakat, MULTICS yazılımı, o zamanki bilgisayarlar için biraz büyük ve hantal kalıyordu.

MULTICS ekibiyle birlikte çalışan ve uzay araştırmalarında kullanılan benzetim *(simulasyon)* yazılımları üretmekte olan **Ken Thompson** hayatından pek memnun değildi. Proje arkadaşları, onun üzerinde çalıştığı programların sistem kaynaklarını çok zorladığından sürekli şikayet ediyorlardı. Bu yüzden, Thompson, sadece başkalarının bilgisayarı kullanmadığı zamanlarda çalışabiliyordu. Bu böyle devam edemezdi. Thompson, çalışmalarını kendisine ait olan eski ve küçük bir DEC PDP-7 bilgisayarında tamamlamaya karar verdi. Ama bu bilgisayarın işletim sistemi de gereksinimlerini karşılamıyordu; bu yüzden kendi istekleri ve gereksinimleri doğrultusunda bir işletim sistemi

geliştirmeye koyuldu. MULTICS'in yararlı bulduğu ve beğendiği özelliklerinin tümünü kullandı. Hatta, o kadar ki, UNIX isminin MULTICS den esinlenildiği; önce UNICS olarak konulduğu, sonradan UNIX'e dönüştürüldüğü **Brian Kernighan** (C Programlama dillini ve UNIX'i yaratan ekibin önemli isimlerinden) tarafından anlatılmaktadır.

1970 yılında UNIX işletim sisteminin ilk sürümü DEC PDP-7 modeli bir bilgisayarda tamamlanmıştı. İşletim sistemi, **programcılar için** yararlı olacak şekilde tasarımlanmış ve özellikle metin işleme yetenekleri *(text processing)* oldukça gelişmişti. 1971 yılında Bell Labs şirketi UNIX işletim sistemini, yeni metin işleme sistemlerinde kullanılacak standart olarak kabul etti. 1972 Haziran ayında gelindiğinde, artık dünyada 10 kadar bilgisayar UNIX işletim sistemi ile çalışmaktaydı. Bu arada, Dennis Ritchie ve Brian Kernighan, **C programlama dili** üzerindeki çalışmalarını büyük ölçüde tamamlamışlardı. 1973 yılında, UNIX işletim sistemi, C programlama diliyle baştan yazıldı. Böylece bilgisayar tarihinin **'yüksek seviyeli bir dil ile yazılmış olan ve donanımdan bağımsız'** ilk işletim sistemi ortaya çıkmış oldu.

1974 yılından başlayarak, AT&T şirketi, bu yeni işletim sistemininin kaynak programlarını, başta Columbia Üniversitesi olmak üzere bir çok üniversite ve yüksek okula ÜCRETSİZ olarak dağıttı. UNIX işletim sisteminin önlenemez yükselişi başlamıştı (aslında önlemek isteyen olduğunu da sanmıyorum).

1975 yılına gelindiğinde, AT&T, UNIX Sürüm 6'yı kullanmaktaydı ve artık UNIX kullanmak isteyenler, küçük de olsa bir lisans ücreti ödemek zorundaydılar. UNIX, standart bir C kütüphanesi ile birlikte dağıtılmaya başlandı. Böylece; C dili, UNIX işletim sistemi için yazılım geliştirmek isteyenlerin öğrenmesi gereken bir dil olarak yaygınlaştı.

1977 yılında, Berkeley Üniversitesi, UNIX üzerindeki birikimlerini ilgilenenlere **1BSD : 1st Berkeley Software Distribution** adlı bir ürün olarak dağıtmaya başladı.

1978 yılında Bill Joy ve Özalp Babaoğlu (University of California-Berkeley'de yüksek lisans öğrencisi) UNIX işletim sistemine sanal bellek özelliğini *eklediler (virtual memory).* Artık UNIX tam bir işletim sistemi olmuştu. *(Ref: Unix Administration Guide for System V, Rebecca Thomas, ISBN 0-13-942889-5).*

1979 yılında, AT&T yedinci sürümü piyasaya çıkardı. UNIX'in yaratıcılarından Ken Thompson'un Berkeley Üniversitesi'nde ders vermeye başlamasıyla AT&T ve Berkeley ekipleri UNIX'i hızla geliştirmeye başladılar. Sonunda, ABD Savunma Bakanlığı'na bağlı DARPA *(Defence Advanced Research Projects Agency : İleri Savunma Araştırma Projeleri)* bölümü, UNIX için bir bütçe ayırmaya karar verdi.

1979'da UNIX artık iyice yaygınlaşmıştı. Üniversite yıllarında UNIX öğrenen, kullanan ve beğenen öğrenciler UNIX'i sanayiye taşımaya ve donanım üreticileri, tasarım aşamalarında UNIX işletim sistemini de göz önünde bulundurma zorunluluğunu hissetmeye başladılar.

1980 yılı sonunda, büyük bilgisayar üreticilerinin hepsi, hiç değilse bazı modellerinde, UNIX kullanmaya başladılar.

Günümüzde (1995) Hewlett-Packard, DEC (Digital Equipment Corporation), IBM, Unisys, Cray Research, SONY, Motorola, NCR, SUN Microsystems gibi devler, UNIX İşletim Sistemi'ni standart olarak desteklemektedir. Kişisel bilgisayarın devi Microsoft'un ve Santa Cruz Oprerations'un (SCO) UNIX'i PC dünyasına taşımasıyla da yayılım tamamlanmış oldu. Bugün, UNIX kullanılan bilgisayar sayısı tam olarak bilinmemekle birlikte, bu sayının milyonlarla ifade edileceği kesindir.

UNIX Geleneği

Çok geniş bir araştırmacı kitlesi tarafından geliştirilmesine rağmen, UNIX, ilk tasarımlandığı günlerdeki özelliklerinden pek uzaklaşmamıştır. Bunun en önemli nedeni, bu araştırmacıların yazılı olmayan geleneklere bağlı kalmış olmalarıdır. Belki de UNIX, başarısını bu gelenekselleşmeye borçludur. (Japon'ların ekonomik mucizesinin de geleneklere bağlılık olduğu söylenmez mi?) Örneğin, dizinlerdeki dosyaların detaylı listesini veren **ls** komutunun 100 Megabyte'dan büyük dosyalarda ortaya çıkan hatası hala düzeltilmemektedir.

Her ne kadar çok tutucu bir tablo çizmiş olsamda, 1990'lı yıllarda iki ayrı UNIX ekolü oluştuğundan söz etmek gerekmektedir: Berkeley Üniversitesinin yürüttüğü **BSD** ekolü ve AT&T şirketinin yürüttüğü AT&T UNIX (**SVR4** : System 5 Release 4) ekolü. Bu iki tip UNIX, kullanıcıları açısından pek önemli farklılıklar göstermese de, sistem yöneticileri açısından çok farklıdırlar. 1992 yılından başlayarak AT&T UNIX'i geliştiren ekipler, BSD UNIX'in üstün özelliklerini AT&T UNIX ile birleştirerek SVR4 UNIX'i ortaya çıkardılar ve BSD ekolüne göre önemli bir üstünlük kazandılar.

UNIX'i UNIX Yapan Özellikler

Belki bazı noktalar tekrar edilmiş olacak ama, UNIX'i UNIX yapan özellikleri bir kez daha sıralamak istiyorum. Kitabın okunması sırasında ve daha önemlisi UNIX İşletim Sistemi'ni kullanırken yararlı olacağı inancındayım.

- ✔ **UNIX 'çok kullanıcılı' bir işletim sistemidir.** Kullanıldığı bilgisayarın bir anda birden fazla kişi tarafından kullanılmasını; daha doğrusu paylaşılmasını sağlayabilmektedir.

- ✔ **UNIX 'çok iş düzeni'ni sağlayan bir işletim sistemidir.** Kullanıcıların, herbirinin, aynı anda birden fazla iş yapmalarına olanak sağlar.

- ✔ **UNIX, donanımdan bağımsızdır.** Hangi bilgisayar üzerinde kullanılırsa kullanılsın, kullanıcılarına göründüğü şekli aynıdır. Öğrendikleriniz kalıcıdır.

- ✔ **UNIX iyi tasarımlanmıştır.** Teknolojideki gelişmelere kolaylıkla uyum sağladığı ve sağlayacağı kanıtlanmıştır.

- ✔ **UNIX, bir işletim sistemi standardı olarak kabul edilmiştir.** Bu sayede farklı marka ve model bilgisayarlar birbirleriyle uyumlu kılınabilmektedir. Son günlerde sıkça sözü edilen 'Bilgi Süper Otoyolu' *(Information Super Highway : Internet)* bu sayede oluşabilmiştir.

UNIX'le Tanışma

UNIX işletim sistemi ile çalışan bir bilgisayarı kullanabilmek için sahip olmanız gereken üç şey vardır :

a) UNIX altında çalışan bir bilgisayara bağlı bir **TERMİNAL'e** (ekran+klavye) erişim yetkisi,
b) UNIX altında çalışan bu bilgisayara erişim hakkınızın anahtarı olan '**kullanıcı hesabınız**' *(user account),*
c) Eğer yeni başlıyorsanız; bol miktarda sabır.

Bu üç özelliğe sahip olduğunuzu varsayarak devam edelim.

Terminalinizi açınız (eğer terminal olarak kullandığınız ekran ve klavye, bilgisayarın ana ekran ve zaten açık olması gerekir.)

Bir kaç saniye içinde ekranda

```
login :
```

mesajını görmeniz gerekir. (Bazı terminallerde bu mesajı görebilmek için bir kaç kez ENTER (ya da RETURN) tuşuna basmanız gerekebilir).

Bu mesaj, bilgisayarın, daha doğrusu UNIX'in, kendinizi tanıtmanızı istediğini belirtmektedir. Her UNIX kullanıcısının bir adı olmalıdır. Bu ad, kullanıcılara **sistem yöneticisi** görevini üstlenmiş olan bilgisayar uzmanları tarafından verilir. Bu mesaja yanıt olarak klavyeden kullanıcı adınızı girmeniz ve ENTER tuşuna basmanız gerekir. Kendi adınızı veya rastgele bir ad girmenizin bir yararı olmayacaktır. UNIX, sadece daha önceden kendisine tanıtılmış olan kullanıcı isimlerini kabul edecektir. Eğer bir kullanıcı adınız yoksa daha fazla vakit kaybetmeden sistem yöneticisini bulup, size bir kullanıcı adı vermesini isteyiniz.

Neyse, geçerli bir kullanıcı adınız olduğunu varsayarak devam edelim...

```
login :ayfer
```

ENTER tuşuna basmanızla birlikte

```
Password :
```

mesajıyla şifrenizi girmeniz istenecektir. Kullanıcı olarak bilgisayara erişiminiz bir şifre ile korunmamışsa, yani sizin için henüz bir şifre girilmemişse, bu mesajı görmezsiniz. Şifreniz yoksa ve bunun özel bir nedeni yoksa, ilk fırsatta kendinize bir şifre seçip, bunu UNIX'e bildirmenizi öneririm. Bu işlem için kullanmanız gereken komut **'passwd'** komutudur. (**passwd** komutunu bir kaç sayfa sonra anlatacağım).

Eğer şifreniz varsa, siz klavyeden bu şifreyi girerken bastığınız tuşlar ekranda görünmeyecektir. (Siz farkında olmadan arkanızdan sizi gözleyenler varsa şifrenizi görmesinler diye...)

UNIX işletim sisteminde **büyük harf - küçük harf** farkı ÇOK önemlidir. **Ayfer**, **AYFER** ve **ayfer** farklı kullanıcı adlarıdır. Aynı fark, şifrelerde de söz konusudur. UNIX geleneği hep küçük harf kullanmanızı (şifreniz hariç) gerektirir.

Doğru şifreyi girdiğinizde (eğer şifre varsa tabii) ekranınızdaki görüntü

```
login : ayfer

Password :

ABC Bilgisayar sistemine hos geldiniz.
Sistem, 12/1/1995 gunu saat 17:00 da bakim icin
kapatilacaktir.

$ _
```

veya

```
login : ayfer

Password :

ABC Bilgisayar sistemine hos geldiniz.
Sistem, 12/1/1995 gunu saat 17:00 da bakim icin
kapatilacaktir.

% _
```

veya

```
login : ayfer

Password :

ABC Bilgisayar sistemine hos geldiniz.
Sistem, 12/1/1995 gunu saat 17:00 da bakim icin
kapatilacaktir.

abc:/home/ayfer $ _
```

gibi olacaktır.

Bu ekranlardaki

```
ABC Bilgisayar sistemine hos geldiniz.
Sistem, 12/1/1995 gunu saat 17:00 da bakim icin
kapatilacaktir.
```

satırları, sistem yöneticisinin kullanıcılara bir mesajıdır (**günün mesajı** *: message of the day*). Sistemdeki yenilikler, kullanıcılara haberler ve duyurular genellikle bu satırlarda yer alır; o nedenle bu mesajları okuma alışkanlığını edinmenizi öneririm.

En son satırlarda yer alabilecek olan

```
$
%
abc:/home/ayfer %
```

satırlarıysa, UNIX'in sizden komut almaya hazır olduğunu belirten '**hazır işareti**''dir ***(prompt)***.

Bu hazır işaretlerinde, UNIX'in sizden komut almaya hazır olduğundan başka çok önemli bir bilgi daha vardır. Bu bilgi, **%** veya **$** karakterleridir. Şimdi sıkı durun, hazır işaretinizde **%** görüyorsanız kullanacağınız **kabuk** (shell) ***Bourne Shell*** 'dir, **$** görüyorsanız ***C Shell*** 'dir. (Sabırlı olmanız gerektiği konusunda uyarmıştım....)

Kabuk (**Shell**) kavramı, UNIX kullanıcılarının iyi anlaması gereken bir kavramdır. Bu noktada MS-DOS işletim sistemi ile bir benzerlik kurmak istiyorum.

MS-DOS'daki **C:\>** benzeri bir hazır işaretinin karşısına yazacağınız komutu irdeleyen, yapılmasını istediğiniz işe ait programı belleğe yükleyen, gerekli parametreleri bu programa aktaran, işletim sisteminin bir parçası COMMAND.COM isimli programdır.

UNIX işletim sisteminde de, aynı şekilde, kullanıcının klavyeden yazacağı komutu irdeleyen, kullanıcının ne yapılmasını istediğini çözümleyen ve bu işin yapılabilmesi için gerekli programları belleğe yükleyen, komut parametrelerinin bu programlara aktarılmasını sağlayan bir program vardır. Bu programların genel adı ***kabuk*** (***shell***) sözcüğüdür. MS-DOS işletim sisteminden farklı olarak, UNIX'de, kullanıcının tercihine bağlı olarak kullanabileceği birden fazla ***komut yorumlayıcısı*** (kabuk = shell) vardır. Bu kabuklara örnek olarak

sh	Bourne Shell	S.R. Bourne	AT&T
csh	C Shell	Bill Joy	Berkeley
ksh	Korn Shell	David Korn	AT&T
bash	Bourne Again Shell		
tcsh	Geliştirilmiş csh		

gösterilebilir.

'Yeni kullanıcılar için şimdilik bu kadar bilgi yeter' deyip devam edelim.

Eğer kullanmakta olduğunuz kabuğun (sistem yöneticisinin sizin için uygun gördüğü kabuk) hangisi olduğunu kesin olarak öğrenmek istiyorsanız

```
% cat /etc/passwd | grep ayfer
```

ayfer sözcüğü yerine kendi kullanıcı adınızı yazmayı unutmayınız!

komutunu yazınız. Göreceğiniz

```
ayfer:As@cX*as:1234:200:Ugur Ayfer:/home/ayfer:/bin/csh
```

benzeri bir satırın en sonuna bakınız. Burada göreceğiniz kabuk programının adı, aradığınız yanıttır.

Eğer kullandığınız UNIX bilgisayarı bir SUN iş istasyonuysa ve verdiğiniz bu komuta yukarıdaki örneğe uygun bir yanıt alamazsanız, bir de

```
% ypcat passwd ¦ grep ayfer
```

komutunu deneyiniz. Gerek duyarsanız sistem yöneticisinden yardım isteyebilirsiniz.

Kabuk Programının Adı	Kabuk Tipi
/bin/csh	C Shell
/bin/sh	Bourne Shell
/bin/ksh	Korn Shell
/bin/bash	Bourne Again Shell
/bin/tcsh	T C Shell

Kullandığınız kabuk programı hangisi olursa olsun, temel UNIX kuralları değişmeksizin geçerli olacaktır. Yeni başlayanların, eğer mümkünse, **csh** kabuk programını kullanmalarını öneririm. Bu kitapta göreceğiniz örneklerin büyük çoğunluğu **csh** için verilecektir.

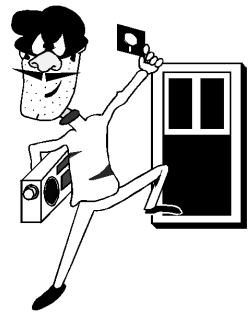
Hangisi olursa olsun; UNIX kabuk programları, MS-DOS işletim sisteminin komut yorumlayıcısı olan COMMAND.COM'la karşılaştırılamayacak kadar gelişmiş ve yeteneklidirler. (Tabii bir o kadar da karmaşık!).

UNIX işletim sistemi ile yapmakta olduğunuz işi tamamladığınızda ve terminalin başından ayrılacağınız zaman

```
% logout
```

komutunu vermeyi unutmamalısınız.

Bu komut, UNIX ile bağlantınızı kesecektir; ve terminal bir sonraki kullanıcıyı bekleme konumuna geçecektir. (`login :`)



UNIX işletim sisteminde **BİR** bilgisayarı paylaşan kullanıcı**LAR** söz konusudur. Bu durumda kullanıcıların kayıtlı bilgilerini birbirlerine karşı korumak gereklidir. Bir sabah işe geldiğinizde tüm kayıtlı bilgilerinizin kaybolduğunu düşünebiliyor musunuz?

Kullanıcıların kayıtlı bilgilerinin yanısıra, işletim sistemi, kendisini de hatalı komutlara ve kötü niyetli kullanıcılara karşı korumak zorundadır. Bu koruma mekanizmasının temelinde **kullanıcı adı** ve **şifresi** yer almaktadır. Her UNIX kullanıcısı şifresini iyi korumak zorundadır. Şifrenizi belki iyi koruyor olabilirsiniz; ancak **logout** komutunu vermeden terminalinizin başından kalkarsanız, arkanızdan terminalin önüne oturan birisi sizin kişiliğinizle UNIX'e vereceği komutlarla bilerek ya da bilmeyerek kayıtlı dosyalarınıza zarar verebilir.

Bir UNIX bilgisayarıyla işiniz bittiğinde **logout** komutunu kullanarak bilgisayarla bağlantınızı kesmelisiniz. Ancak, **logout** etmeniz, bilgisayarı da kapatabileceğiniz anlamına gelmez.

Lütfen; ama lütfen, UNIX işletim sistemi ile çalışan bir bilgisayarı işiniz bittiğinde **küüt diye kapatmayınız**. Bir UNIX bilgisayarının sağlıklı bir şekilde kapatılabilmesi için bir dizi törensel işlem yapılması gerekir. Eğer bu işlemleri yapmadan kapatırsanız, bilgisayarı bir daha açamayabilirsiniz; hatta kayıtlı tüm veri ve programları kaybedebilirsiniz.

Bir UNIX bilgisayarın kapatılması için gereken törensel işlemler, bu kitabın 'Sistem Yöneticisine' başlıklı bölümünde anlatılacaktır.

Isınma Hareketleri

Kullanıcı ile UNIX İşletim Sistemi arasındaki tüm haberleşme **kabuk** (shell) programı aracılığı ile yürütülmektedir. Klavyeden yazacağınız her komut, kullanmakta olduğunuz kabuk programı tarafından yorumlanmaya çalışılacaktır. Eğer kullanmakta olduğunuz kabuk için anlamı olmayan komutlar yazacak olursanız, beklemediğiniz hata mesajları ile karşılaşabilirsiniz. Bu bölümdeki örnekler **csh** kabuğu için hazırlanmıştır. Eğer kullandığınız kabuk Bourne Shell (**sh**) ise (hazır işaretinizin sonunda **$** karakteri varsa), klavyeden

```
$ /bin/csh
```

komutunu vererek **C Shell kabuğuna** geçmeyi deneyiniz. Eğer bir hata mesajı almazsanız ve hazır işaretiniz sonunda **%** karakteri olan bir diziye dönüşürse başardınız demektir.

```
login : ayfer
Password :

        Günün mesajları

$ /bin/csh
abc:/home/ayfer %            c-shell'e geçiş başarılı...
```

Eski kabuğunuza dönmek istediğinizdeyse, **Ctrl-D** ye basmalı veya **exit** komutunu vermelisiniz.

Ben Kimim?

İlk bakışta çok anlamlı değilmiş gibi görünen bu soru UNIX dünyasında zaman zaman sorulması gereken bir sorudur. Eğer kullandığınız UNIX bilgisayarı büyük bir bilgisayar ağının bir parçasıysa ve siz bu ağ üzerinden bir çok bilgisayara ulaşabiliyorsanız ve bu değişik bilgisayarlardaki kullanıcı isimleriniz *(user-id)* farklıysa; uzun çalışma seansları sırasında, o anda geçerli olan kullanıcı kimliğinizi şaşırabilirsiniz.

Hemen

```
% whoami                    BSD UNIX'lerde
% who am i                  SV5R4 UNIX'lerde
```

komutunu verip, UNIX'in sizi o anda hangi kimlikle tanıdığını öğrenebilirsiniz. Özellikle sistem yöneticileri, zaman zaman başka kullanıcıların kimliğine bürünme gereksinimi duyarlar (bu işi **su** - *switch user* komutuyla yaparlar). Bir o - bir bu kullanıcı kimliğine büründüklerinde de bazen şaşırmalar olur.

Böyle bir durumda hemen **whoami** komutunu vererek o andaki kimliklerini öğrenebilirler.

Başka Kimler Var?

UNIX işletim sistemi altında çalışan bilgisayarların, bir anda birden fazla kullanıcı tarafından kullanılabileceğini belirtmiştim. İsterseniz, şu anda bilgisayarı sizden başka kullanan kimse var mı, onu öğrenelim. Bunun için vermeniz gereken komut:

```
% who
```

```
abc:/home/ayfer % who
ayfer           tty01       Jan 12   15:12
hakan           tty03       Jan 12   10:09
root       console    Jan 11   23:40
abc:/home/ayfer %
```

Yukarıdaki örneğe göre, şu anda bilgisayarı paylaşan 3 kişi olduğunuz anlaşılıyor. Diğer ortaklarınızın isimleri ***hakan*** ve ***root***. Hakan 3 numaralı terminalin, **root** ise ana terminalin *(konsol)* başında oturuyor. **hakan** 12 Ocak günü saat 10:09 da **login** etmiş; **root** ise bir gün önce gece yarısına doğru çalışmaya başlamış. Eğer, **root** gerçekten dün geceden beri çalışıyorsa mesele yok; ama eğer gece eve gitmiş ve giderken **logout** komutunu vermemişse önemli bir güvenlik hatası yapmış demektir.

UNIX kullanıcılarının isimleri genellikle kullanıcıların gerçek kimliklerini yansıtacak şekilde seçilir. Sistem yöneticisi; bir kullanıcı tanıtımı yaparken, kullanıcı hesap ismi yanısıra, bu kullanıcının bilgisayardaki kaynaklara erişim yetkilerini de tanımlar.

Ancak, UNIX işletim sisteminde adı hiç bir zaman değişmeyen ÖZEL bir kullanıcı vardır. Bu kullanıcının adı, ***root*** sözcüğüdür. Adı **root** olan kullanıcı HER ŞEYİ YAPMAYA YETKİLİDİR. İstediği dosyayı siler, yaratır, yerini ve içeriğini değiştirir vs. vs. Bu kullanıcıya "süper kullanıcı" *(super user)* adı da verilir.

Eğer bir UNIX bilgisayarına **root** kullanıcı olarak erişme hakkınız varsa (yani **root** şifresini biliyorsanız), gerekmedikçe bu isimle **login** etmeyiniz. Yapacağınız hatalar sisteminizi çalışmaz hale getirebilir. UNIX işletim sistemi, **root** isimli kullanıcının yaptığı işi çok iyi bildiğini varsayıp, hiç bir uyarıda bulunmaksızın verilen komutları yerine getirir. (**her şeyi sil** komutu dahil!)

Arayan Soran Var mı?

UNIX işletim sisteminde, kullanıcılar arasında elektronik posta haberleşmesinin yapılmasını sağlayan **e-mail** *(electronic mail)* yazılımı standarttır. Kullanıcılar birbirlerine göndermek istedikleri mesajları (**elektronik posta** veya kısaca **mektup**)

```
% mail
```

komutunun yardımıyla yazarlar, gönderirler ve kendilerine gelen mektupları gene bu komutla okurlar.

mail komutunu parametresiz olarak kullandığınızda :

```
abc:/home/ayfer % mail
You have no mail.          (Mektubunuz yok.)
  veya
No messages
```

yanıtları yanısıra, size gönderilmiş mektup(lar) varsa:

```
Mail ver 4 Thu Jan 31 12:54 EST 1995 Type ? for help
"/usr/mail/ayfer":3 messages 2 new
U  1 cil@bilkent   Fri Jan 12 14:32 23/567 Yeni uygu.
N  2 tayfun@abc    Fri Jan 12 15:34 34/762 Onemli
N  3 kerem@abc     Fri Jan 23 09:12 45/947 SUNOS4.1
&
```

gibi size gelen mektupların bir listesini görebilirsiniz. Bu mektup listesinde, size mektubu gönderen kullanıcının adı, mektubun konusuna ilişkin kısa bir not ve mektup sıra numarası yer alır. Tamamını okumak istediğiniz mektubun numarasını girdiğinizde elektronik mektubunuzun tamamını okuyabilirsiniz. Okumak istediğiniz mektuplar bitince, **x** tuşuna basarak **mail** programından çıkabilirsiniz. Bu komutun kullanımı ile ilgili detayları daha sonraki bölümlerde anlatacağım.

Siz sisteme bağlı değilken, adresinize (kullanıcı adınıza) bir mektup gelirse, ilk **login** edişinizde, UNIX sizi

```
You have new mail
```

diye uyaracaktır.

Bu uyarıyı gördüğünüzde **mail** komutunu kullanarak gelen mektuplara bakabilirsiniz; bu mektupların sizi ilgilendirmediğini ya da başka birisini de ilgilendirdiğini düşünüyorsanız, mektubu başka bir adrese yönlendirebilirsiniz, mektubu saklayabilirsiniz ya da çöpe atabilirsiniz. (Eğer Internet bağlantınız varsa, her gün bir sürü çöpe atılacak mektup alacağınızdan emin olabilirsiniz).

Şifrenizi Değiştirmek İstediğinizde...

UNIX altında çalışan bir bilgisayara sizin adınızı (yani **kullanıcı adınızı** demek istiyorum) kullanarak ulaşabilen herkes, size gelen elektronik mektupları da okuyabilir. Başkalarının size ait dosyaları ve elektronik mektupları okumasını istemiyorsanız, UNIX'in şifre mekanizmasından yararlanmanız gerekecektir. Bilgisayara erişim şifrenizi (password) değiştirmek istediğinizde

```
% passwd
```

komutunu kullanmalısınız. Eğer şifreli bir kullanıcı adı ile çalışıyorsanız, yeni şifre verebilmek için o anda geçerli olan şifreyi bilmeniz gerekecektir.

```
abc:/home/ayfer % passwd

Changing old password for  ayfer
Old password :                    eski şifreyi veriniz
New password :                    yeni şifreyi giriniz
Retype new password :             yeni şifreyi bir kez
                                  daha giriniz.
```

Şifreyi iki kez girmenizin istenmesi oldukça mantıklı değil mi? Klavyeden yazarken ekranda göremeyeceğiniz bir şifreyi hatalı yazarsanız, bir daha bu sisteme **login** etmeniz olanaksız hale gelecektir.

Şifrenizi seçerken bazı noktalara dikkat etmelisiniz!

Seçtiğiniz şifre, sizin tarafınızdan kolayca hatırlanacak; ancak başkaları tarafından kolayca tahmin edilemeyecek bir karakter dizisi olmalıdır. Eşinizin veya çocuğunuzun adı, soyadınız, arabanızın plakası, doğum tarihiniz şifre olarak kullanılması sakıncalı olan dizilerdir. Şifre olarak çok karmaşık diziler seçip, sonra da bu şifreyi unutmamak için bir kenara yazmak da oldukça sık yapılan güvenlik hatalarındandır.

Şifrenizi seçerken, mümkün olduğunca harf ve sayıları karıştırınız. Daha iyisi hem büyük, hem küçük harfleri bir arada kullanınız.

Şifreniz ne çok uzun, ne de çok kısa olsun. 6 - 8 karakterlik diziler hem kolay hatırlanır, hem de klavyeden yazılırken pek hata yapılmaz.

ayfer	Çok kötü bir şifre, hemen tahmin edilir.
AyfeR-1995	Hiç fena değil.
123456	Çok ciddiyetsiz, üstelik klavyeden yazarken kolayca izlenir.
aBcDeF	Fena değil ama çok kişi buna benzer şifre kullandığı için kötü niyetli kişilerce ilk denenen kalıplardandır.
x1e34TQ?w/&1+	Harika bir şifre, ama siz hatırlayabilecek misiniz?

Sisteme **login** ettiğinizde, UNIX genellikle bir önceki **login** seansınızın hangi tarihte gerçekleştiğini size hatırlatır. Bu hatırlatmaya her **login** edişinizde dikkatlice bakmanızı öneririm. Bu mesaj sayesinde, sizin adınızı kullanarak sisteme ulaşan birileri varsa, durumu farkedebilirsiniz. Böyle bir durumdan şüphelendiğiniz anda şifrenizi değiştiriniz. Daha da iyisi, şifrenizi en az ayda bir kez değiştiriniz. Nitekim, bazı sistem yöneticileri, kullanıcıları, şifrelerini belirli sıklıklarda değiştirmeye otomatik olarak zorlarlar. (*Password aging* kavramı).

Şifreniz, sistem yöneticisinin, diğer adıyla **root** kullanıcısının, dosyalarınıza bakmasına engel olamaz. **root** herşeyi olduğu gibi, mektuplarınızı ve diğer dosyalarınızı da okumaya yetkilidir.

İmdaaaaat !..

UNIX işletim sisteminde kullanılabilecek yüzlerce komut vardır. Seyrek kullanılan komutların genel yapılarını ve parametrelerinin hepsini hatırlamak pek kolay olmadığı için; UNIX işletim sistemi, tüm komutlarının kullanım kılavuzlarını standart olarak size sunmaktadır. Bir komutun nasıl kullanılacağını öğrenmek ya da hatırlamak istediğinizde

```
% man komut-adi                                    (manual)
```

komutunu vermeniz, *komut-adı* adlı komutun kullanım kılavuzu sayfalarının ekranınızda görüntülenmesini sağlayacaktır. Örneğin, **passwd** komutunun nasıl kullanılacağını merak ederseniz

```
% man passwd
```

mail komutunun nasıl kullanılacağını hatırlamak içinse

```
% man mail
```

komutlarını kullanabilirsiniz.

İşiniz bittiğinde...

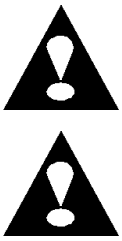
Bilgisayarla işiniz bittiğinde, terminalinizin başından ayrılmadan önce

```
% logout
```

komutunu veriniz.

Sistem yöneticiniz (ya da siz) aksini belirtmediyseniz, (**.cshrc Dosyasıyla** ilgili bölümde **ignoreeof** parametresine bakınız) **Ctrl-D** tuşuna basarak da sistemle bağlantınızı kesebilirsiniz. Ctrl-D aslında bağlantı kesme komutu değil, o anda aktif olan komut yorumlayıcınızı (kabuk) öldürme komutudur. Eğer öldürdüğünüz kabuk, yegane kabuğunuzsa sistemle bağlantınız kesilir; yok

ikinci ya da üçüncü kabuğunuzsa, bir önceki kabuğunuza dönersiniz. Bu kavram biraz karışık geldiyse aldırmayın, zamanla açıklığa kavuşacaktır.



Bir UNIX bilgisayarın başında işiniz bittiğinde **logout** komutunu kullanarak bilgisayarla bağlantınızı kesmelisiniz. Ancak, **logout** etmeniz, bilgisayarı da kapatabileceğiniz anlamına gelmez.

Lütfen; ama lütfen, UNIX işletim sistemi ile çalışan bir bilgisayarı işiniz bittiğinde **küüt diye kapatmayınız**. Bir UNIX bilgisayarının sağlıklı bir şekilde kapatılabilmesi için bir dizi törensel işlem yapılması gerekir. Eğer bu işlemleri yapmadan kapatırsanız, bilgisayarı bir daha açamayabilirsiniz; hatta kayıtlı tüm veri ve programları kaybedebilirsiniz.

Bir UNIX bilgisayarın kapatılması için gereken törensel işlemler, bu kitabın 'Sistem Yöneticisine' başlıklı bölümünde anlatılacaktır.

UNIX Dosya Yapısı (UNIX File System)

Tüm bilgisayar işletim sistemlerinin olduğu gibi, UNIX'in de en temel amacı kullanıcıların verilerini ve programlarını bilgisayar ortamında düzenli bir şekilde saklamalarına yardımcı olmaktır. UNIX işletim sisteminde tüm veriler, programlar ve herbiri aslında bir program olan komutlar, **dosya**'larda *(file);* dosyalarsa dizinlerde (**dizin** : *directory*) gruplanmış olarak saklanır.

UNIX dosya yapısını anlatırken okuyucunun MS-DOS işletim sistemine aşina olduğunu varsayacağım ve bu nedenle sık sık MS-DOS'la karşılaştırmalar yapacağım. Bu arada da sık sık UNIX'in MS-DOS'a karşı ezici üstünlüğünü vurgulamış olacağım. Bu nedenle MS-DOS hayranlarından şimdiden özür dilerim.

UNIX işletim sisteminde dosya isimlerine ilişkin kurallar oldukça esnektir.

En başta, MS-DOS'daki gibi 8 karakterden oluşan isim ve 3 karakterden oluşan uzantı *(extension)* kavramı yoktur. Dosya isimleri, UNIX uyarlamasına bağlı olarak değişmekle birlikte 255 karaktere kadar uzunlukta olabilir (bu uzunlukta dosya isimlerini kim hatırlayıp klavyeden yazacaksa...).

Nokta (**.**) karakterinin özel bir anlamı yoktur. Dosya adı içinde istediğiniz kadar nokta kullanabilirsiniz. Ancak, nokta ile başlayan dosya isimleri bir anlamda özeldir; adı nokta ile başlayan dosyalar yarı gizli dosyalardır. Özellikle belirtmedikçe, dosya isimleri listelerinde bu tür dosyaları göremezsiniz.

Dosya isimlerinde büyük harf-küçük harf ayırımı VARDIR. **ayfer.mektup**, **Ayfer.Mektup** ve **AYFER.MEKTUP** tamamen farklı dosya isimleridir.

Bir kaç örnek vermek gerekirse :

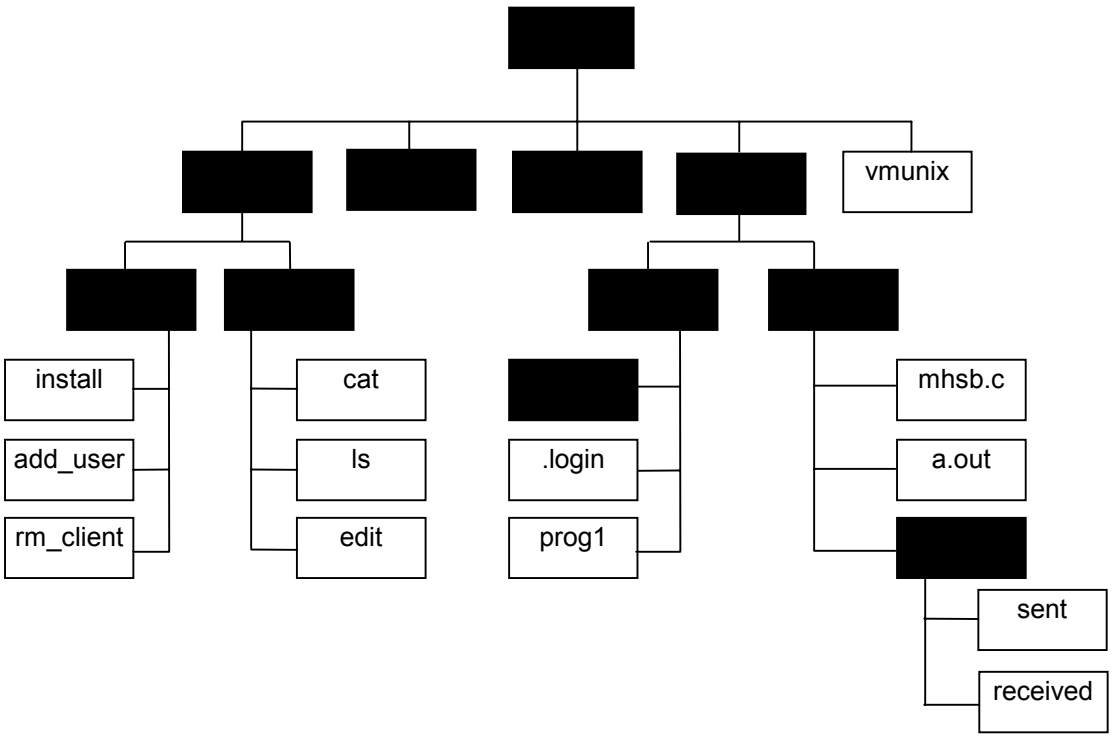
`ayfer.mektuplar`	*Geçerli bir dosya adı,*
`a1`	*Geçerli bir dosya adı,*
`1a`	*Geçerli bir dosya adı,*
`1-a`	*Geçerli bir dosya adı,*
`muhasebe_1995_mizan`	*Geçerli bir dosya adı,*
`Sinanin.Muhasebe.Programi`	*Geçerli bir dosya adı,*
`.login.eski`	*Geçerli bir dosya adı,*
`lotus.exe`	*Geçerli bir dosya adı,*
`prog1.com`	*Geçerli bir dosya adı,*

Dosya isimleriyle, dosyaların program olup olmaması arasında bir ilişki yoktur.

Örneğin, **lotus.exe** isimli bir dosyanın, bir program dosyası olması gerekmez. Bir dosyanın program dosyası olup olmadığını isminden anlayamazsınız. Program dosyalarının diğer dosyalardan nasıl ayırdedileceğini bir kaç sayfa sonra anlatacağım.

UNIX, MS-DOS'dan tanıdığınız hiyerarşik dosya-dizin yapısını kullanmaktadır. En üst düzeyde bir **root** dizini ve bunun altında istendiği gibi yerleştirilmiş olan dosya ve alt-dizinler ile gene bu alt-dizinler altında yerleştirilmiş dosyalar ve gene alt-dizinler...

Şematik olarak göstermek gerekirse :



Dikkat ederseniz, MS-DOS dosya yapısından farklı olarak '**root**' dizininin adı **** *(back-slash)* değil, normal **/** *(slash)* karakteridir. Aynı şekilde, bir dosyanın dizinler arasındaki yerini tanımlarken, MS-DOS'daki **** karakteri yerine **/** karakteri kullanılır. Bunu örneklerle göstermek gerekirse; yukarıdaki dosya-dizin yapısında yer alan bazı dosyaların tam isimleri şöyle yazılır :

UNIX	MS-DOS
/usr/bin/cat	C:\USR\BIN\CAT
/home	\home
/home/sina/Mail/sent	\HOME\SINA\MAIL\SENT
/vmunix	C:\VMUNIX

Her horoz kendi çöplüğünde...

UNIX işletim sisteminde, her kullanıcının kendisine ait bir '**kullanıcı dizini**' (UNIX terminolojisinde : ***home directory***) vardır. Bu dizin, kullanıcının sisteme tanıtımı sırasında, sistem yöneticisi tarafından yaratılır. Her kullanıcının kendi '**kullanıcı dizini**'nde sınırsız yetkileri vardır. Bu dizin altında istediği gibi dosya ve alt dizinler yaratır, bunları siler, isimlerini ve içeriklerini değiştirir vs. vs.

Her kullanıcının kendi dizinindeki bu yetkileri, başka kullanıcıların dizinleri üzerinde yoktur. Bir başka deyişle, **ayfer** isimli kullanıcı, **sina** isimli kullanıcının dizinindeki dosyaları silemez, değiştiremez, **sina** izin vermedikçe okuyamaz; hatta varlığından bile haberdar olamaz.

Sisteme **login** eden her kullanıcı, çalışma dizini, kendisine ait kullanıcı dizini olacak şekilde çalışmaya başlar. Sistem yöneticileri, kullanıcı dizinlerini, genellikle **/home** dizini altına açtıkları dizinler olarak düzenlediklerinden (tipik bir UNIX geleneği) **ayfer** isimli kullanıcının **login** ettiğinde kendini **/home/ayfer** adlı dizinde bulması doğaldır.

```
login : ayfer
Password :
 ... Günün mesajları ...
abc:/home/ayfer %
```

Bu örnekteki '**abc**', kullandığınız UNIX bilgisayarının adıdır. Eğer bilgisayarınız bir bilgisayar ağına bağlıysa, bu ad çok önemli olacaktır.

root isimli kullanıcı, kime ait olursa olsun, tüm dosya ve dizinler üzerinde, bu dosya ve dizinler sanki kendisininmiş gibi tam yetkilidir. İsterse siler, isterse değiştirir.

Lütfen **root** isimli kullanıcıyla, dizin yapısının en üst düzeyindeki **root** dizinini ('/') karıştırmayınız. Her iki kavram için de aynı sözcüğün kullanılmasının nedeni, her şeye yetkili olan ***root*** isimli super kullanıcının, kendisine ait olan 'kullanıcı dizini' nin tüm dizin yapısını temsil eden / dizini olmasıdır.

Her ne kadar, UNIX sizi kendi kullanıcı dizininize yerleştirdiyse de, bu yerleşim mutlak değildir. İsterseniz **cd** komutu ile çalışma dizininizi *(default directory)* değiştirebilirsiniz.

```
abc:/home/ayfer % cd  /usr/etc
abc:/usr/etc %
```

cd komutunu kullanarak çalışma dizininizi değiştirdiğinizde, **hazır işaretinde *(prompt)*** yeni çalışma dizinin adının yer alması bir tercihtir. Eğer hazır işaretinizde çalışma dizininizi göremiyorsanız sistem yöneticisine başvurunuz. Bu işi kendiniz halletmek istiyorsanız, kitabın ileri bölümlerinde bu işin nasıl yapılacağını anlatacağım.

Neredeyim?

Eğer kullandığınız sistem, hazır işareti içinde size bulunduğunuz çalışma dizinini göstermiyorsa

```
% pwd                                    (print working directory)
```

komutunu kullanarak çalışma dizininizi öğrenebilirsiniz.

Örneğin,

```
login : ayfer
Password :
   ...
Günün mesajları
   ...
% pwd
/home/ayfer
%
```

Ne var ne yok?

Çok doğal olarak, bulunduğunuz dizinde yer alan dosyaların listesini görmek isteyeceksiniz.

Kullanacağınız komut en basit haliyle :

```
% ls                                                    (list)
```

Hemen bir örnek vereyim :

```
/home/ayfer % cd /
/ % ls
bin         export      lost+found        tmp
boot        home        mnt               usr
dev         kadb        sbin              var
etc         lib         sys               vmunix
/ %
```

Pek açıklayıcı olmadı galiba; değil mi? Hangisinin dosya, hangisinin dizin olduğu belli değil; oysa MS-DOS bu farkı **<DIR>** sembolü ile belirtirdi (!).

Daha açıklayıcı bir liste isterseniz **ls** komutunun yanına **-F** seçeneğini koyabilirsiniz (Dikkat! F büyük harf olmalı) :

```
/home/ayfer % cd /
/ % ls -F
bin/        export/     lost+found/ tmp/
boot        home/       mnt/              usr/
dev/        kadb*       sbin/             var/
etc/        lib/        sys/              vmunix*
/ %
```

Bu listede dizinler, isimlerinin sonuna yerleştirilen **"/"** karakterleriyle; program veya komut dosyalarıysa **"*"** ile belirtilmiş olarak karşınıza çıkacaktır. Herhangi bir eki olmayan isimlerse, program dosyası veya dizin olmayan, diğer tip dosyalara aittir.

Bazı isimlerin sonunda **"@"** işareti göreceksiniz. Bu işaretin anlamını açıklamak için henüz biraz erken; ama şu kadarını söyleyebilirim : **"@"** işaretli dosya veya dizinler, aslında orada olmayan dosya ve dizinleri belirler. Nasıl ama? Esrarengiz değil mi? Var ama aslında yok...

Bu liste her zaman alfabetik sırada ve dosya isimlerinin izin verdiği ölçüde birden fazla sütun halinde dökülecektir. Bu listeye önce ilk sütunu, sonra diğer sütunları görecek şekilde bakmaya alışmalısınız.

Dosyalar ve dizinler hakkında daha detaylı bilgi istiyorsanız aşağıdaki komutu denemelisiniz. **ls** komutunun bu formunu MUTLAKA deneyiniz ve bu form ile alacağınız listenin nasıl yorumlandığını lütfen ÇOK ÇOK İYİ ANLAYINIZ. UNIX mantığını iyi kavrayabilmeniz açısından buradan başlayarak anlatacaklarım oldukça önemlidir.

```
% ls -l                                                    (long list)
```

```
/home/ayfer % cd /
/ % ls -l
total 3166

lrwxrwxrwx  1 root         7 Jan  12 12:09 bin -> /usr/bin
-r--r--r--  1 root    110912 Jan  12 12:11 boot
drwxr-sr-x  2 bin       7680 Jan  12 12:23 dev
drwxr-sr-x  7 bin       1536 Jan  15 08:45 etc
drwxr-sr-x  4 root       512 Feb   1 11:56 export
drwxr-xr-x  5 root       512 Mar  23 09:03 home
-rwxr-xr-x  1 root    239783 Feb  09 13:34 kadb
lrwxrwxrwx  1 root         7 Mar  01 18:23 lib -> /usr/lib
drwxr-xr-x  2 root      8192 Jun  15 23:09 lost+found
drwxr-sr-x  2 bin        512 Mar  01 20:09 mnt
drwxr-sr-x  2 bin        512 Mar  09 08:59 sbin
lrwxrwxrwx  1 root        13 Jan  24 07:45 sys -> /usr/kvm/sys
drwxrwsrwt  2 bin        512 Feb  24 09:56 tmp
drwxr-xr-x 20 root       512 Nov  23 16:08 usr
drwxr-xr-x 11 root       512 Nov  23 16:11 var
-rwxr-xr-x  1 root   1101191 Jan  11 09:35 vmunix
/ %
```

Bu ayrıntılı liste, inanamayacağınız kadar çok bilgi içermektedir. Bu aşamada bütün detaylara girmeyeceğim; sadece satırlardan birini örnek olarak ele alıp, bir fikir verecek şekilde kısaca açıklayacağım.

```
-rwxr-xr-x  1 root    239783 Feb  09 13:34 kadb
```

`-rwxr-xr-x` : Bu satırın bir dosyayla ilgili olduğunu (en baştaki **-** işaretinden anlıyoruz); bu dosyanın sahibinin bu dosyada okuma (**r** : read), yazma (**w** : write) ve çalıştırma (**x** : execute) yetkilerinin olduğunu; diğer kullanıcıların, sadece okuma ve çalıştırma yetkilerinin bulunduğunu; dolayısıyla bu dosyanın bir **program dosyası** olduğunu;

`root` : Bu dosyanın sahibinin **root** isimli kullanıcı olduğunu;

`239783` : Dosyanın uzunluğunun 239,783 byte olduğunu;

`Feb 09 13:34 :` Dosyanın en son 9 Şubat saat 13:34 de değişikliğe uğradığını;

`kadb :` Dosyanın adının **kadb** olduğunu göstermektedir.

Dizinler içinse, bu **ls** satırı biraz farklıdır :

```
drwxr-xr-x 20 root        512 Nov  23 16:08 usr
```

En baştaki **d** harfi, listenin bu satırının bir dizine ait olduğunu göstermektedir. Dosya uzunluğu yerinde yazılı olan sayıysa, dizinlerde pek anlamlı değildir; daha doğrusu anlamı konumuzun tamamen dışındadır.

Her satırdaki `rwxr-xr-x` benzeri kalıplarda gördüğünüz kodlar, kullanıcıların dosya (ya da dizin) üzerindeki erişim yetkilerini tanımlamaktadır. Programın ilerleyen saatlerinde bu erişim yetkileri konusu detaylı olarak ele alınacaktır. Bizden ayrılmayın.

```
% man ls
```

man ls komutunu verdiğinizde, aşağıda göreceğiniz uzun açıklamalar ekranınıza listelenecektir. Bu açıklamalar, kullandığınız UNIX işletim sistemine ait kullanım kılavuzunun **ls** komutu ile ilgili bölümlerinin aynısıdır. **man** komutunu verdiğinizde, listelenecek satırlar bir ekran sayfasından fazlaysa, birinci sayfanın listelenmesi tamamlanınca, ekranın sol alt tarafında

```
--- more ---
```

işareti göreceksiniz. Bu mesaj; listelenen açıklamaların devamı olduğunu; bu sayfayı okumayı tamamlayınca klavyeden bir komut vererek listenin devamını görmenizin mümkün olduğunu belirtmektedir.

`--- more ---` un karşısına :

bo□luk	tuşuna (Space Bar) basarsanız, bir sonraki sayfa,
RETURN	tuşuna basarsanız, bir sonraki satır,
b	(küçük b) tuşuna basarsanız bir ÖNCEKİ sayfa listelenir. Bu geri gitme özelliği her UNIX uyarlamasında (örneğin SCO UNIX) çalışmaz.

Şimdi, **ls** komutunun detaylarını öğrenmek için **man ls** komutunu bir deneyiniz.

```
abc:/home/ayfer % man ls
Reformatting page.  Wait... done

LS(1V)                    USER COMMANDS                    LS(1V)

NAME
     ls - list the contents of a directory

SYNOPSIS
     ls [ -aAcCdfFgilLqrRstu1 ] filename ...


     /usr/5bin/ls [ -abcCdfFgilLmnopqrRstux ] filename ...

AVAILABILITY
     The System V version of this command is available  with  the
     System  V software installation option.  Refer to Installing
     SunOS  4.1  for  information  on  how  to  install  optional
     software.

DESCRIPTION
     For each filename which is a directory, ls  lists  the  con-
     tents  of  the directory; for each filename which is a file,
     ls repeats its name and any other information requested.  By
     default, the output is sorted alphabetically.  When no argu-
     ment is  given,  the  current  directory  is  listed.   When
     several  arguments are given, the arguments are first sorted
     appropriately,  but  file  arguments  are  processed  before
     directories and their contents.

     In order to determine output formats for the -C, -x, and  -m
     options, /usr/5bin/ls uses an environment variable, COLUMNS,
     to determine the number of character positions available  on
     one  output line.  If this variable is not set, the terminfo
     database is used to determine the number of  columns,  based
     on  the environment variable TERM.  If this information can-
     not be obtained, 80 columns are assumed.

  Permissions Field
     The mode printed under the -l option contains 10  characters
     interpreted as follows.  If the first character is:

          d  entry is a directory;
          b  entry is a block-type special file;
          c  entry is a character-type special file;
          l  entry is a symbolic link;
          p  entry is a FIFO (also known as "named pipe") special
             file;
          s  entry is an AF_UNIX address family socket, or
          -  entry is a plain file.

     The next 9 characters are interpreted as three sets of three
     bits  each.   The first set refers to owner permissions; the
     next refers to permissions to others in the same user-group;
     and  the  last  refers  to  all others.  Within each set the
     three characters indicate permission respectively  to  read,
     to write, or to execute the file as a program.  For a direc-
     tory, "execute" permission is interpreted to mean permission
     to  search  the directory.  The permissions are indicated as
     follows:

          r  the file is readable;
          w  the file is writable;
          x  the file is executable;
          -  the indicated permission is not granted.

     The group-execute permission character is given as s if  the
     file  has  the  set-group-id  bit  set;  likewise the owner-
     execute permission character is given as s if the  file  has
     the set-user-id bit set.

     The last character of the mode (normally x or `-') is  t  if
     the 1000 bit of the mode is on.  See chmod(1V) for the mean-
     ing of this mode.  The indications of set-ID and  1000  bits
     of  the  mode  are capitalized (S and T respectively) if the
     corresponding execute permission is not set.

     When the sizes of the files in a  directory  are  listed,  a
     total count of blocks, including indirect blocks is printed.
```

```
OPTIONS
     -a   List all entries; in the absence of this option,
          entries whose names begin with a `.' are not listed
          (except for the super-user, for whom ls, but not
          /usr/5bin/ls, normally prints even files that begin
          with a `.').

     -A   (ls only)  Same as -a, except that `.' and `..' are not
          listed.

     -c   Use time of last edit (or last mode change) for sorting
          or printing.

     -C   Force multi-column output, with entries sorted down the
          columns;  for ls, this is the default when output is to
          a terminal.

     -d   If argument is a directory, list only its name (not its
          contents);  often  used  with -l to get the status of a
          directory.

     -f   Force each argument to be interpreted  as  a  directory
          and  list  the  name  found  in each slot.  This option
          turns off -l, -t, -s, and -r,  and  turns  on  -a;  the
          order  is  the  order  in  which  entries appear in the
          directory.

     -F   Mark directories with a trailing slash  (`/'),  execut-
          able  files  with  a  trailing asterisk (`*'), symbolic
          links  with  a  trailing  at-sign  (`@'),  and  AF_UNIX
          address  family  sockets  with  a  trailing equals sign
          (`=').

     -g   For ls, show the group ownership of the file in a  long
          output.   For  /usr/5bin/ls,  print a long listing, the
          same as -l, except that the owner is not printed.

     -i   For each file, print the i-number in the  first  column
          of the report.

     -l   List in long format,  giving  mode,  number  of  links,
          owner, size in bytes, and time of last modification for
          each file.  If the file is a special file the size
          field  will  instead contain the major and minor device
          numbers.  If the time of last modification  is  greater
          than  six  months ago, it is shown in the format `month
          date year';  files  modified  within  six  months  show
          `month  date time'.  If the file is a symbolic link the
          pathname of the linked-to file is printed  preceded  by
          `->'.  /usr/5bin/ls will print the group in addition to
          the owner.

     -L   If argument is a symbolic link, list the file or direc-
          tory the link references rather than the link itself.

     -q   Display non-graphic  characters  in  filenames  as  the
          character ?; for ls, this is the default when output is
          to a terminal.
     -r   Reverse the order of sort to get reverse alphabetic  or
          oldest first as appropriate.

     -R   Recursively list subdirectories encountered.

     -s   Give size of each file, including any  indirect  blocks
          used  to  map  the  file, in kilobytes (ls) or 512-byte
          blocks (/usr/5bin/ls).

     -t   Sort by time modified  (latest  first)  instead  of  by
          name.

     -u   Use time of last access instead  of  last  modification
          for  sorting (with the -t option) and/or printing (with
          the -l option).

     -1   (ls only) Force one entry per line output format;  this
          is the default when output is not to a terminal.

SYSTEM V OPTIONS
     -b   Force printing of non-graphic characters to be  in  the
          octal \ddd notation.

     -m   Stream output format; the file names are printed  as  a
          list  separated by commas, with as many entries as pos-
          sible printed on a line.

     -n   The same as -l, except that the owner's UID and group's
          GID  numbers  are  printed,  rather than the associated
```

```
          character strings.

     -o   The same as -l, except that the group is not printed.

     -p   Put a slash (`/') after each filename if that file is a
          directory.

     -x   Multi-column output with entries sorted  across  rather
          than down the page.

ENVIRONMENT
     The environment variables  LC_CTYPE,  LANG,  and  LC_default
     control  the  character  classification  throughout  ls.  On
     entry to ls, these environment variables are checked in  the
     following  order:  LC_CTYPE,  LANG,  and LC_default.  When a
     valid value is found, remaining  environment  variables  for
     character  classification  are  ignored.  For example, a new
     setting for LANG does not override the current valid charac-
     ter  classification  rules  of  LC_CTYPE.   When none of the
     values is valid, the shell character classification defaults
     to the POSIX.1 "C" locale.

FILES
     /etc/passwd         to get user ID's for  `ls  -l'  and  `ls
                         -o'.
     /etc/group          to  get  group  ID  for  `ls   -g'   and
                         `/usr/5bin/ls -l'.
     /usr/share/lib/terminfo/*
                         to   get   terminal   information    for
                         /usr/5bin/ls.
SEE ALSO
     chmod(1V)
     The output device is assumed to be 80 columns wide.

     The option setting based on whether the output is a teletype
     is  undesirable  as  `ls -s' is much different than `ls -s |
     lpr'.  On the other hand, not doing this setting would  make
     old shell scripts which used ls almost certain losers.

     None of the above apply to /usr/5bin/ls.

     Unprintable characters in file names may confuse the  colum-
     nar output options.
```

Dosyalar (Files)

Kullanıcıların, bir bilgisayarla yaptıkları çalışmaların meyveleri dosyalardır. Çizim programları kullanarak hazırladığınız çizimler, bir sonraki çalışma adımı için dosyalarda (disk, disket veya teypde) saklanır. Yazdığınız programlar dosyalar olarak saklanır ve kullanılır. Muhasebe kayıtları yıl boyunca dosyalarda biriktirilir. Bütün bu dosyaları yaratıp yaşatabilmeniz için gerekli desteği, size, kullandığınız bilgisayarın işletim sistemi verir. Dosyalarınızı düzenlemek, kopyalarını çıkarmak, yedeklerini almak, günlük işlerinizin önemli bir parçası olacaktır; bu nedenle UNIX'in dosya kavramını uzun uzun anlatmak istiyorum.

UNIX'de dosyalar, aynı MS-DOS'da olduğu gibi, ilgili oldukları uygulamaya göre düzenlenmiş dizinlerde *(directory)* saklanır. UNIX'in çok kullanıcılı bir işletim sistemi olmasından dolayı, diskin veya disklerin kullanıcılar arasında paylaşılması, dosyaların konuları yanısıra, kullanıcıların kimliklerine göre de gruplanmasını gerektirir. (Hatırlarsanız, her kullanıcının bir 'kullanıcı dizini' olduğundan daha önce söz etmiştim.)

Dosya ve dizinlerin kullanıcılar arasında paylaşılmasından dolayı, bir kullanıcıya ait dosyaların ve dizinlerin bir 'erişim yetkisi' mekanizması ile diğer kullanıcılara karşı korunması gerekmektedir. UNIX'de bu koruma mekanizması, kullanıcıların sisteme tanıtılması sırasında verilen '**kullanıcı ismi**', '**kullanıcı numarası**' ve kullanıcının ait olduğu '**çalışma grubunun numarası**' na dayandırılır. Normal şartlar altında, kullanıcıların kendilerine ait 'kullanıcı numarası'nı bilmelerine gerek yoktur, kullanıcı ismi ve kullanıcı numarası arasındaki bağlantı, işletim sistemi tarafından otomatik olarak sağlanır.

Dosya Yaratma

Dosya yaratmanın bir çok yöntemi vardır :

- Bir editör kullanarak metin dosyaları;
- Bir muhasebe programı kullanarak, yeni bir yıl için veri dosyaları;
- Bir bilgisayar destekli tasarım program paketi kullanarak çizim dosyaları;
- Yazdığınız bir C programını derleyerek amaç program *(object code)* ve makina kodu *(machine code, executable code)* dosyaları;
- Eski bir dosyanın kopyasını çıkararak yeni bir dosya;
- Teypten ya da CDROM'dan diske kopyalamak yoluyla diskte bir dosya;
- Program çıktılarını yönlendirerek (I/O Redirection) döküm dosyaları
- vs. vs. vs.

yaratabilirsiniz.

Biz en basitinden başlayalım :

cat komutu

UNIX'de çok sık kullanılan, çok işlevli bir komuttur. Bu işlevlerden bir tanesi,, MS-DOS daki TYPE komutu ile aynıdır.

En basit kullanım formu :

```
% cat  dosya_adi                                    (concatenate)
```

şeklindedir. Bu formda kullanıldığı zaman, **dosya_adı** adlı dosyayı ekrana (daha doğrusu, UNIX diliyle **STANDART ÇIKTI BİRİMİ'**ne : *Standard Output*) gönderir. Standart çıktı birimi genellikle ekran olduğu için, bir dosyayı ekrana listelemek için kullanılır.

Denemek için

```
% cat  /etc/motd
```

komutunu verebilirsiniz.

Dosya yaratmak için kullanacağımız form ise biraz daha farklı....

```
% cat > yenidosya
```

Bu formda kullanıldığında, **cat** komutu, **STANDART GİRDİ BİRİMİ'**nden (klavyeden, *Standard Input*) aldığı bilgileri, **yenidosya** isimli bir dosyaya yönlendirecektir (bir başka deyişle kopyalayacaktır).

Şimdi isterseniz **dosya1** isimli ilk küçük dosyamızı yaratalım:

```
% cat > dosya1
```

komutunu veriniz,

daha sonra imleç *(cursor)* yeni satırın başına geldiğinde, dosyanın içinde yer almasını istediğiniz satırları giriniz; örneğin

```
abc:/home/ayfer % cat > dosya1
Bu bizim ilk deneme dosyamiz.
Icinde sadece iki satir var.
^D
abc:/home/ayfer %
```

Girmek istediğiniz satırlar tamamlanınca, imleç satır başındayken Ctrl ve D tuşlarına birlikte basarak *(EOF : End of file karakteri)* standart giriş biriminizde dosya sonuna gelindiğini belirtin.

Dosya adı verirken dizin adı belirtmediğiniz için, **dosya1** adlı dosya çalışma dizininizde yaratılır. Herhangi bir hata mesajı almadıysanız; dosya problemsiz yaratıldı demektir. Eğer **dosya1** in yaratılıp yaratılmadığını kontrol etmek isterseniz iki yöntem önerebilirim :

```
% cat  dosya1
```

Dikkat ederseniz ">" karakteri yok...

ve

```
% ls
```

Birinci komut (**cat**), **dosya1** dosyasının içine yazdıklarınızı ekrana görüntüleyecek, böylece yaratma işleminin tamamlanıp tamamlanmadığını çok sağlam bir şekilde kontrol etmiş olacaksınız. İkinci komutlaysa (**ls**) sadece dosyanızın adını, uzunluğunu, ne zaman yaratıldığını ve sahibinin kim olduğunu göreceksiniz.

Bence iki yöntemi de deneyiniz.

Eğer **cat** komutunu parametresiz olarak verirseniz,
komut pek de anlamlı olmayan bir iş yapmaya; standart giriş biriminden okuyup, standart çıkış birimine kopyalamaya başlayacaktır. Yani klavyeden (standart giriş birimi) bastığınız her tuş, standart çıkış birimine (ekran) kopyalanacaktır.

Yanlışlıkla düşebileceğiniz bu durumdan kurtulmak için, imleç satır başındayken **Ctrl-D** tuşuna basınız. Bu hareketiniz kopyalama işini sona erdirecektir. Bu işlem, diskteki dosyalarınızı hiç bir şekilde etkilemez.

Küçük bir varyasyonla **cat** komutunu ilginç bir iş yapmak için de kullanabiliriz.

```
abc:/home/ayfer % cat >> dosya1
Bu sat□rlar, dosya1 dosyasinin arkasina eklenecek.
ve dosya1 toplam 4 satir olacak.
^D
abc:/home/ayfer %_

abc:/home/ayfer % cat  dosya1
Bu bizim ilk deneme dosyamiz.
Icinde sadece iki satir var.
Bu sat□rlar, dosya1 dosyasinin arkasina eklenecek.
ve dosya1 toplam 4 satir olacak.
abc:/home/ayfer %_
```

cat komutunu bir de şu şekilde deneyiniz :

```
cat > /dosya1
Bu bizim ilk deneme dosyamiz.
Icinde sadece iki satir var.
^D
Access denied.
```

Evet, tahmin ettiğim gibi bir hata mesajı aldınız (*Access denied* : Bu işi yapmaya yetkiniz yok!).

Sebebi açık... **dosya1** isimli dosyayı **root** (**/**) dizininin hemen altında yaratmaya çalıştınız ve sizin bu dizine kayıt yapmaya yetkiniz olmaması da çok doğal.

Bu örneği **root** kullanıcı olarak yapmış olsaydınız (lütfen denemeyiniz!), böyle bir mesajla karşılaşmayacaktınız.

cp komutu (copy kelimesinden türemiştir)

Bu komutun ne işe yaradığını söylemeye gerek olduğunu sanmıyorum; ama nasıl kullanıldığı önemli...

En basit formuyla

```
% cp  dosya_adi_1   dosya_adi_2          (copy)
```

dosya_adı_1 isimli dosyayı **dosya_adı_2** isimli dosyaya kopyalayacaktır.

Eğer **dosya_adı_2** isimli dosya yoksa, yaratılacaktır (tabii bu dosyanın yer alacağı dizinde dosya yaratmaya yetkiniz varsa...) Eğer bu isimde bir dosya eskiden varsa, üzerine kopyalama yapılacak ve eski içeriği bozulacaktır. Böyle bir durumda, **eski bir dosyanın üzerine kayıt yapmak üzere olduğunuz konusunda uyarılmayacaksınız!** Dikkatli olmanız gerekir.

Eğer dikkatinize güvenmiyorsanız, **cp** komutunu

```
% cp -i  dosya_adi_1   dosya_adi_2
```

formunda kullanın**. -i** parametresi *(interactive),* eski bir dosyanın üzerine kayıt yapılması durumunda kullanıcının **Overwrite?** mesajı ile uyarılmasını ve ancak **y** yanıtı verilirse devam edilmesini sağlar.

-i parametresini kullanmayı unutmaktan korkuyorsanız, kitabın **alias** komutu ile ilgili bölümünü okuyunuz; bu bölümde çeşitli UNIX komutlarını kalıcı olarak değiştirmenin, hatta kendinize özgü UNIX komutları yaratmanın yollarını bulacaksınız.

Bir başka form :

```
% cp  dosya_adi   dizin_adi
```

dosya_adı isimli dosyayı, **dizin_adı** isimli dizinin altına kopyalar. İsterseniz **-i** opsiyonunu gene kullanabilirsiniz.

Bu formla ilk form arasında görünüş olarak hiç bir fark yoktur. İkinci parametreyle verilen isim bir dizine aitse, verdiğiniz komut ikinci form olarak kabul edilir ve birinci dosya bu dizinin altına kopyalanır. İkinci parametreyle belirtilen isimde bir dosya varsa, ya da bu isimde hiç bir şey (dosya veya dizin) yoksa; ilk form kabul edilerek ilk parametredeki dosyanın kopyası çıkarılır.

Hemen bir örnek :

Çalışma dizininizde

```
-rw-rw-rw-  1  root   918 Jan 12 12:09  ocak1995
-rw-rw-rw-  1  root   918 Jan 12 12:09  subat1995
```

dosyaları varken,

```
% cp ocak1995 veriler
```

komutu verirseniz, çalışma dizininizdeki yeni dosyal listesi

```
-rw-rw-rw-  1  root   918 Jan 12 12:09  ocak1995
-rw-rw-rw-  1  root   918 Jan 12 12:09  subat1995
-rw-rw-rw-  1  root   918 Jan 14 17:43  veriler
```

şekline dönüşür.

Eğer komutu vermeden önce; çalışma dizininizin görünüşü

```
-rw-rw-rw-  1  root   918 Jan 12 12:09  ocak1995
-rw-rw-rw-  1  root   918 Jan 12 12:09  subat1995
drwxrwxrwx  1  root   918 Jan 14 17:43  veriler
```

gibi idiyse; (**veriler** isimli bir dizin bulunduğuna dikkatinizi çekerim) **cp** komutundan sonra, **ocak1995** dosyası, **veriler** dizininin altına kopyalanmış olacaktır.

Bir başka form :

```
% cp  dosya1 dosya2 dosya3 ....  dizin_adi
```

dosya1, dosya2, dosya3, vs. vs. isimli dosyaları, **dizin_adı** isimli dizinin altına kopyalar. İsterseniz **-i** parametresini kullanabilirsiniz.

```
% cp  dosya1 dosya2 ... dosyaN  dizin_ad□
```

formunu kullanığınızda, **dizin_adı** adlı dizin bulunamazsa, garip şeyler olacaktır. Önce **dosya1** isimli dosya dizin_adı isimli bir dosyaya kopyalanacaktır. Sonra **dosya2** isimli dosya gene **dizin_adı** isimli dosyanın üzerine kopyalanacaktır. Daha sonra da bu işlem **dosya3** ve varsa diğer dosyalar için tekrarlanacaktır. Sonunda, **dosyaN** in bir kopyası **dizin_adı** isimli bir dosyaya çıkarılmış olacaktır.

MS-DOS kullan□c□lar□... Dikkat!

UNIX **cp** komutunda, kopyalamanın **nereden nereye** yapılacağını mutlaka açıkça belirtmelisiniz. Yani,

```
% cp  /dizin1/dosya1
```

şeklinde bir komut **kullanamazsınız**. Bu şekilde yazacağınız bir komut, **/dizin1**'in altındaki **dosya1** isimli dosyayı, çalışma dizinine kopyala anlamına **gelmez**; hatalı bir komuttur.

Dizin Kopyalama

UNIX'de, **dizin** kopyalamak için de **cp** komutu kullanılır; ancak özel bir parametreyle birlikte..

Dizin kopyalamak için kullanılan form :

```
% cp  -r  dizin1  dizin2          dizin kopyalama
```

şeklindedir. Bu formda verilen kopyalama komutu; varsa, dizinlerin alt-dizinlerinin de kopyalanmasını sağlar. (MS-DOS'da **/S** parametresiyle kullanılan **XCOPY** komutu gibi...)

MS-DOS kullanıcıları, el alışkanlığıyla sık sık, **cp** yerine **copy** yazacaklardır. Sizinde böyle bir sıkıntınız olur ve kurtulmak isterseniz **alias** komutuyla

ilgili bölümünü okuyunuz; bu komutla **copy** ismiyle kendi kopyalama komutunuzu tanımlayabilirsiniz. **cp** komutu üzerine bir kaç gelişmiş örnek vererek bu konuyu geçmek istiyorum :

`% cp /etc/motd /tmp/motd2`	**/etc** isimli dizinin altındaki **motd** isimli dosyayı, **/tmp** isimli dizinin altına, adını **motd2** olarak değiştirerek kopyala.
`% cp /etc/motd /tmp`	**/etc** isimli dizinin altındaki **motd** isimli dosyayı, **/tmp** isimli dizinin altına, adını değiştirmeden kopyala.
`% cp /etc/motd .` Nokta	**/etc** isimli dizinin altındaki **motd** isimli dosyayı **çalışma dizinine** (bir başka deyişle, **buraya**) kopyala. (Tek nokta, **'burası'** anlamındadır; aynı MS-DOS'daki gibi).
`% cp /home/ayfer/prg1 ..` İki nokta	**/home** isimli dizinin altındaki **ayfer** dizininin altındaki **prg1** isimli dosyayı çalışma dizininin bir üstündeki dizine kopyala. (İki nokta yanyana, 'bir üst dizin' anlamındadır; aynı MS-DOS'daki gibi).
Çarpıcı bir örnek: `% rcp abc:/vmunix temel:/sakla` **rcp** : Remote Copy	**abc** bilgisayarının **root** dizinindeki **vmunix** isimli dosyayı **temel** isimli bir başka bilgisayardaki, varsa **sakla** isimli dizinin altına; böyle bir dizin yoksa, **root** dizininin altına **sakla** ismiyle kopyala. (Bu komutu verebilmeniz için, **abc** ve **temel** isimli bilgisayarların bir bilgisayar ağı ile birbirlerine bağlı olmaları ve sizin iksine dei erişim hakkınız olması gerekmektedir.) Bu ve buna benzer komutlara daha sonra ayrıntılı olarak değineceğim.

Dizin Yaratma

UNIX'de, dizin yaratmak için **mkdir** komutu kullanılır.

Formu basittir :

```
% mkdir  dizin                                  (make directory)
% mkdir eski_dizin/yeni_dizin
```

gibi...

Doğal olarak, yalnızca yetkiniz olan yerlerde dizin yaratabilirsiniz...

Unutmayınız !

Komutlar hakkında daha detaylı bilgiye gereksinim duyarsanız; örneğin parametrelerini hatırlayamazsanız; **man** komutu her zaman kullanımınıza hazırdır.

Sırf alışkanlık kazanmak amacıyla hemen şimdi

```
% man  mkdir
```

komutunu vermeye ne dersiniz?

Dosya Silme

Artık diskte yer almasını istemediğiniz dosyaları silmek için kullanacağınız komut

```
% rm  dosya                                    (remove)
% rm  dosya1 dosya2 dosya3 ... dosyaN
```

formlarındadır.

Bir seferde (tek komutta), farklı dizinlerde yer alan dosyaları da silebilirsiniz.

```
% rm /dizin1/dosya1 /baska_dizin/dosya2 ...
```

Eğer dosyalar silinmeden önce onaylamak istiyorsanız **-i** parametresini kullanabilirsiniz:

```
% rm  -i  /dizin1/dosya1 /baska_dizin/dosya2 ...
```

komut formunu kullandığınızda, silinecek her dosya için teker teker

```
remove ?
```

sorusu sorulacak ve sadece **y** yanıtını verdiğiniz dosyalar silinecektir.

Dizin Silme

Artık diskte yer almasını istemediğiniz dizinleri, altlarındaki dosya ve alt dizinleriyle birlikte silmek için kullanacağınız komut

```
% rm  -r  dizin                                    (remove)
% rm  -r  dizin1 dizin2 dizin3  ... dizinN
```

formlarındadır.

Bir seferde, farklı dizinlerde yer alan dizinleri de silebilirsiniz.

```
% rm /dizin1/alt_dizin1 /baska_dizin/dizin2 ...
```

Eğer dizinlerin silinmeden önce tarafınızdan onaylanmasını istiyorsanız **-i** parametresini kullanabilirsiniz:

```
% rm  -i  /dizin1/alt_dizin1 /baska_dizin/dizin2 ...
```

komut formunu kullandığınızda, silinecek her dizin için teker teker

remove ?

sorusu sorulacak ve sadece **y** yanıtını verdiğiniz dizinler silinecektir.

Çok Önemli !

UNIX işletim sisteminde UNDELETE görevini yerine getirecek bir program ya da komut yoktur. Sildiğiniz dosya ve dizinler, bir daha geri getirilemeyecek şekilde silinir. Bu nedenle rm komutunu kullanmadan önce iyi düşünmelisiniz.

Dosya/Dizin Adı Değiştirme

Bu iş için kullanacağınız komut **mv** (move) komutudur. Dosya ve dizin ismi değiştirmek için kullanılan formu basittir :

```
% mv  eski_dosya_ismi  yeni_dosya_ismi        (move)
% mv  eski_dizin_ismi  yeni_dizin_ismi
```

Bir isim değişikliği yapmak istediğinizde, doğal olarak, söz konusu dosya veya dizinin yer aldığı dizinde, yeni isimde bir dosya ya da dizin bulunmamalıdır.

Dosya/Dizin Yeri Değiştirme

Bu iş için kullanacağınız komut gene **mv** *(move)* komutudur. Dizin yeri değiştirmek için kullanılan özel **-R** parametresine dikkatinizi çekerim.

```
% mv  eskiyeri\dosya  yeniyeri\dosya          move
% mv -R eskiyeri\dizin  yeniyeri\dizin
```

Bir yer değişikliği yapmak istediğinizde, doğal olarak, söz konusu dosya veya dizinin yer alacağı yeni dizinde, aynı isimde bir dosya ya da dizin bulunmamalıdır.

Çalışma Dizinini Değiştirme

Aynı MS-DOS işletim sisteminde olduğu gibi, bir dosyanın adını verirken, dosyanın yer aldığı dizini tam olarak belirtmezseniz, dosyanın o andaki çalışma dizininizde bulunduğu varsayılır. Çalışma dizinini değiştirmek için kullanılan komut

```
% cd  yeni_calisma_dizini                (change directory)
```

aynı MS-DOS'daki CD komutu gibidir.

Örnekler vermek gerekirse :

`% cd /home/ayfer/proje1`	Pek açıklama gerektirmiyor sanırım...
`% cd ../proje2`	Bir üstteki dizinin altındaki **proje2** isimli dizine geç.

`% cd ../../mektuplar`	İki üst düzeydeki dizinin altındaki **mektuplar** isimli dizine geç.
Kullanışlı bir olanak : `% cd ~omer`	Kullanıcı adı **omer** olan kullanıcının kullanıcı dizinine geç. (**omer**'in **home** dizini)
Hoş bir olanak daha... `% cd` 	Her nerede olursan ol, şu anda geçerli olan kullanıcı adına ait **home** dizinine geç. (Yuvaya dönüş!)

Dizinler arasında gidip gelirken, zaman zaman kaybolmanız doğaldır. Özellikle **prompt**'unuz (hazır işaretiniz) çalışma dizininiz hakkında bilgi vermiyorsa...

Kaybolduğunuzda, **pwd** komutu ile *(print working directory)* o andaki çalışma dizininizin hangi dizin olduğunu öğrene-bilirsiniz.

Buraya kadar temel bir kaç UNIX komutundan; sık sık da yetkilerden söz ettik. Sanırım şu **yetki** meselesini biraz daha açmanın zamanı geldi...