

Sistem Programlama

Ders 3

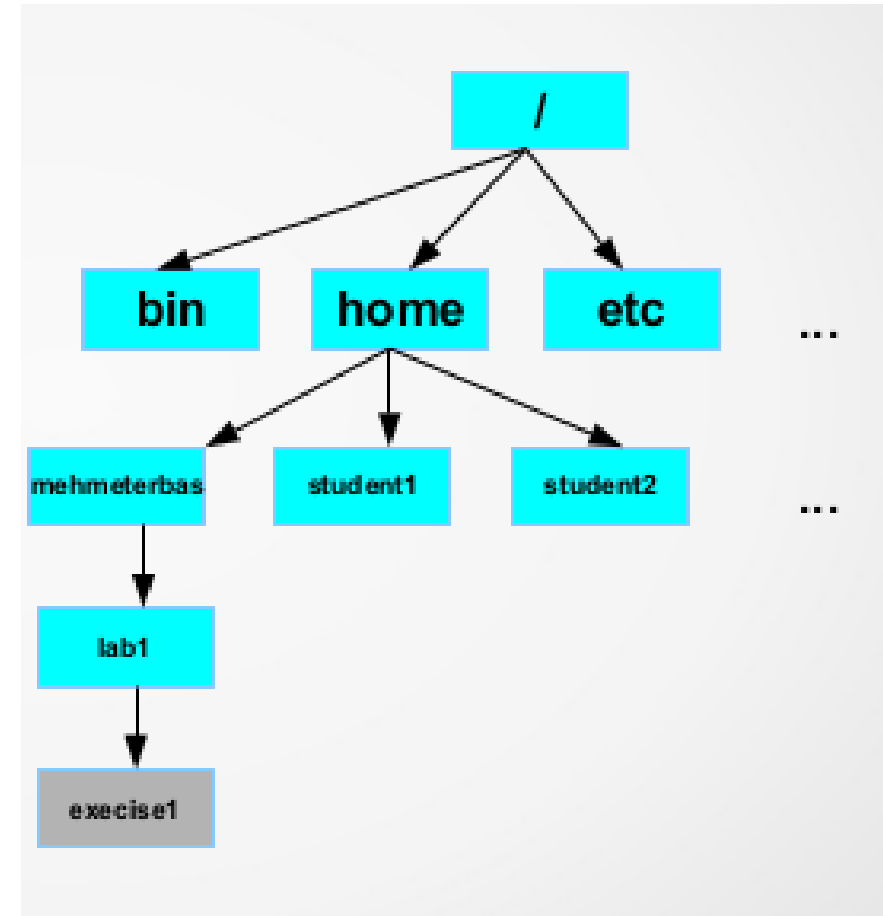
Doç. Dr. Mehmet Dinçer Erbaş
Bolu Abant İzzet Baysal Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Unix sistemine genel bakış

- Dosya sistemi
 - Unix sistemi dosyaları ve klasörleri hiyerarşik olarak düzenler.
 - Her şey kök klasörü ile başlar: /
 - Klasör, diğer dosyaları içerebilen özel bir dosya tipidir.
 - Bir klasörü ismi olan ve içerdiği diğer dosyalara dair bilgiler içeren özel bir yapı olarak düşünebiliriz.
 - Dosyanın içerdiği özelliklerden bazıları şunlardır:
 - Dosya tipi (normal dosya, klasör gibi).
 - Dosyanın büyüklüğü
 - Dosyanın sahibi
 - Dosya üzerindeki izinler
 - Dosyanın son değişme zamanı
 - stat ve fstat komutları dosyanın özellikleri ile ilgili bilgi verir.

Dosyalar ve klasörler

- Bir klasör için dosya isimleri bulunur.
 - Bir dosya ismi içerisinde / karakteri ve null karakteri bulunamaz.
- Yeni bir klasör oluşturulduğunda iki dosya otomatik olarak oluşturulur.
 - . (dot) ve .. (dot-dot)
- Günümüzde neredeyse tüm Unix sistemleri 255 karakterlik dosya isimlerine izin verir.



Dosyalar ve klasörler

- Yoladı
 - Bir veya daha fazla birbirini takip eden dosya ismi, aralarında / karakteri ile, ve başlangıçta opsiyonel olarak / karakteri olan bir metin yoladı (ing: pathname) olarak adlandırılır.
 - / karakteri ile başlayan yoladları mutlak yoladı olarak adlandırılır.
 - `cd /home/mehmeterbas/lab1`
 - Diğerleri ise görece yoladı olarak adlandırılır,
 - `cd mehmeterbas/lab1`

Dosyalar ve klasörler

- Örnek1: ls1.c
- Örnek program hakkında notlar
 - apue.h: Bu dosya ders boyunca kitabımızda göreceğimiz örnekler için gerekli standart kütüphaneleri ve sabit değerleri içerir.
 - Programdaki main fonksiyonunun deklarasyonu ISO C standartlarına uygundur.
 - Program kullanıcıdan bir argüman almakta (argv[1]) ve bu argümanı dosyaları listelenecek klasör adı olarak kullanmaktadır.
 - Klasör işlemleri yapabilmek için opendir, readdir ve closedir fonksiyonlarını kullanıyoruz.

Dosyalar ve klasörler

- Örnek program hakkında notlar
 - opendir fonksiyonu dir yapısında bir işaretçi döner.
 - Daha sonra readdir fonksiyonu ile bir döngü içerisinde klasördeki kayıtları okur.
 - readdir fonksiyonu bütün kayıtları okuduğunda null işaretçi döner ve bu sayede döngü sonlanır.
 - Her kayıttaki dirent yapısında bulunan d_name bilgisi kullanıcıya gösterilir.
 - Ayrıca programda uygun hata mesajları tanımlanmıştır.
 - Program sonlandığında exit fonksiyonu 0 argümanı ile çağırılmıştır.

Dosyalar ve klasörler

- Çalışma klasörü
 - Her işleme ait bir çalışma klasörü mevcuttur.
 - Bu işlem için bütün görece yolları bu klasörden hareketle hedefine gider.
 - Bir işlem chdir fonksiyonunu kullanarak çalışma klasörünü değiştirebilir.
 - doc / memo / joe
 - Bu yoladı joe isminde bir dosyayı hedeflemektedir. Bu dosya memo isimli bir klasördedir, memo klasörü doc isimli bir klasördedir ve doc klasörü çalışma klasöründe bulunmalıdır.
 - / usr / lib / lint
 - Bu yoladı lint isimli bir dosyayı hedeflemektedir. lint dosyası lib isimli bir klasördedir, lib klasörü ise usr klasöründedir. usr klasörü ise işletim sisteminin kök (root) klasöründedir.
 - Giriş dizini
 - Sisteme giriş yaptığımızda çalışma klasörü olarak giriş dizinine gireriz.
 - Giriş dizini parola dosyasında bulunan kaydımızdan öğrenilir.

Girdi ve çıktı işlemleri

- Dosya tanımlayıcıları
 - Dosya tanımlayıcıları kernel tarafından işlemlerin kullanmakta oldukları dosyaları tanımlayabilmek için kullanılan negatif-olmayan tam sayılardır.
 - İşlemler tarafından açılan her dosyanın bir dosya tanımlayıcısı bulunur.
- Bununla birlikte kabuk tarafından açılan üç farklı dosya tanımlayıcısı mevcuttur. Bunlar:
 - Standart girdi, standart çıktı, standart hata.
- Daha önce belirttiğimiz gibi kabukların birçoğu otomatik açılan bu dosyaları yönlendirmek için gereken yöntemleri sağlar.
 - `ls > file.list`

Girdi ve çıktı işlemler

- Tamponsuz (unbuffered) girdi çıktı işlemler open, read, write, lseek ve close fonksiyonları ile yapılır.
 - Bu dosyalar üzerinde işlem yapabilmek için dosya tanımlayıcıları kullanılır.
- Örnek2: mycat.c

Girdi ve çıktı işlemleri

- mycat.c
 - `STDIN_FILENO` ve `STDOUT_FILENO` POSIX standartlarının bir parçasıdır.
 - Bu sabit değerler `<unistd.h>` dosyasında tanımlanmıştır.
 - Bu iki sabit aslında 0 ve 1 değerleri olarak tanımlanmıştır.
 - `read` fonksiyonu okunan byte sayısını döner
 - Okunan byte sayısı `write` fonksiyonunda kullanılıyor.
 - Girdi tanımlandığında `read` fonksiyonu 0 döner ve program sonlanır.
 - Program çalışırken `read` fonksiyonunda bir hata oluşursa -1 değeri döner.

Girdi ve çıktı işlemleri

- Standart I/O
 - Standart I/O fonksiyonları kullanarak tamponlu (buffered) girdi çıktı işlemleri yapılabilir.
 - Standart I/O fonksiyonu kullanırsanız doğru tampon büyüklüğünü bulmak zorunda kalınmaz.
 - Ayrıca satır satır okumak için özel fonksiyonlar vardır.
 - fgets ==> tam bir satırı okur.
 - Standart I/O fonksiyonlarının en ünlüsü printf fonksiyonudur.
 - Örnek3: getcputc.c
 - getc fonksiyonu birer birer karakterleri okur ve okunan karakter putc tarafından yazılır.
 - En son byte okunduktan sonra getc EOF değeri döner ve bu değer <stdio.h> kütüphanesinde tanımlıdır.
 - Kütüphanenin kullandığı stdin ve stdout sabitleri <stdio.h> kütüphanesinde tanımlıdır.

Programlar ve işlemler

- Program
 - Program disk üzerinde bulunan çalıştırılabilir bir dosyadır.
 - Bir program çalıştığında önce hafızaya yüklenir ve kernel tarafından 6 farklı exec fonksiyonundan biri kullanılarak çalıştırılır.
- İşlemler ve işlem ID
 - Bir program çalıştığında bir işlem oluşur.
 - Bazı işletim sistemlerinde işlem yerine görev kelimesi kullanılır.
 - Unix sisteminde her işlem için ayrı işlem ID verilir. İşlem ID sayıları negatif olmayan tam sayılardır.
- Örnek4: hello.c
- İşlem kontrolü
 - İşlem kontrolü için üç temel fonksiyon kullanılır.
 - fork, exec, waitpid
 - Örnek5: shell1.c

Program ve işlemler

- shell1.c
 - Örnekte işlem kontrol fonksiyonu kullanarak standart girdiden komutları okuyan ve bu komutları çalıştıran bir program görülmektedir.
 - Kabuk benzeri programlar buna benzer şekilde oluşturulur.
 - Standart I / O fonksiyonu olan fgets fonksiyonu standart girdiden satır okumak için kullanılır.
 - Program çalışırken CTRL-D yani dosya sonu karakteri girdiğimizde fgets fonksiyonu NULL işaretleyici döner, döngü sonlanır ve işlem sonlanır.
 - fgetc fonksiyonu ile alınan satırlar yeni satır (newline) karakteri ile sonlanır ve null byte tarafından takip edilir.
 - Alınan satırdaki yeni satır karakteri null byte ile değiştirilir çünkü kullandığımız fonksiyon olan execlp fonksiyonu null byte ile biten bir argüman beklemektedir.

Programlar ve işlemler

- shell1.c
 - fork fonksiyonunu kullanarak yeni bir işlem yaratırız. Yaratılan işlem fork çağrısını yapan işlemin kopyasıdır.
 - Çağırان işlem üst işlem, yeni yaratılan işlem ise alt işlemidir.
 - Fork fonksiyonu üst işleme yeni yaratılan alt işlemin işlem numarasını dönerken, alt işlemde 0 döner.
 - Alt işlem execlp fonksiyonunu kullanarak standart girdiden girilen komutu çalıştırır.
 - Üst işlem alt işlemin sonlanmasını bekler.
 - Bekleme işlemi waitpid fonksiyonu ile yapılır ve bu fonksiyona beklenecek olan alt işlemin işlem numarası verilir.
 - Bu programın en büyük kısıtlaması şu anki versiyonunda komutlara argüman gönderemenin mümkün olmamasıdır.

Programlar ve işlemler

- İşlemcik ve işlemcik numarası
 - Genelde, bir işlem sadece bir kontrol işlemciği içerir.
 - Ancak bazı durumlarda birden fazla işlemi aynı anda yapmak istersek, problemin farklı parçalarını çözen işlemcikler oluşturabiliriz.
 - Bir işlem içerisindeki bütün işlemcikler aşağıdakileri paylaşır.
 - Adres alanı (ing: address space)
 - Dosya tanımlayıcıları
 - Stack (ing: yığın)
 - İşlem ile alakalı özellikler.
 - Her işlemcik kendi yığına sahiptir. Ancak aynı işlem içerisindeki diğer işlemciklerin yığınlarına ulaşabilirler.
 - İşlemciklerin aynı hafızaya erişebildiklerinden dolayı kendi aralarında senkronize olmaları gerekmektedir.

Programlar ve işlemler

- İşlemciler, işlemci numarası ile tanımlanırlar.
 - İşlemci numarası sadece bulunduğu işlem içerisinde anlamlıdır.
- İşlemci numarası kullanarak işlem içerisinde istediğimiz işlemciğe ait komutlar çalıştırabiliriz.
- İşlemciler Unix sistemine çok sonradan eklenmiştir.