

LUENTO 1

582497 Käyttöjärjestelmät, 8 op

Tiina Niklander

Helsingin yliopisto
Tietojenkäsittelytieteen laitos

Kurssin asema

- *Hajautettujen järjestelmien ja tietoliikenteen erikoistumislinjan* maisteriopintojen pakollinen kurssi
- Esitietoina edellytetään:
 - Kandidaatin tutkinnon opinnot, ainakin kurssit
 - Tietokoneen toiminta
 - Rinnakkaisohjelmointi
- Edeltäjiä:
 - Tietokone ja käyttöjärjestelmä
 - Käyttöjärjestelmät I ja II

Kurssin tavoite

- Ymmärtää KJ:n perusrakenteet ja toteutusperiaatteet sekä käyttäjän että toteuttajan näkökulmasta
- Tuntee yleisperiaatteiden lisäksi myös vaihtoehtoisia algoritmeja ja toteutuksia
- Esimerkeinä
Unix = Linux, Windows XP ja Solaris
- Kurssin jälkeen:
 - Osaa kuvata käyttöjärjestelmän osat, niiden toiminnan ja vuorovaikutuksen
 - Pystyy tarvittaessa vaikka lähdekoodin avulla selvittämään miten tietty käyttöjärjestelmä toteuttaa jonkin tietyn toiminnon

Kurssikirja

- Kurssikirja
 - [Stal05] William Stallings, *Operating Systems, 5th. ed.*, Prentice-Hall, 2005
- Oheislukemistoa
 - [Tane01] A.S. Tanenbaum, *Modern Operating Systems, 2nd. ed.*, Prentice-Hall, 2001.
 - [DDC04] H.M. Deitel, P.J. Deitel, D.R. Choffnes, *Operating Systems, 3rd ed.*, Prentice-Hall 2004
 - Mikä tahansa kirja, jonka otsikossa maagiset avainsanat *Operating Systems*

Kurssin rakenne:

- Osio 1: Prosessit ja säikeet
 - Myös käyttöjärjestelmän rakenne ja tehtävät
- Osio 2: Muistinhallinta
 - Erityisesti virtuaalimuisti
 - Myös vuorottaminen
- Osio 3: Siirräntä ja tiedostojärjestelmät
- Osio 4: Hajautus ja tietoturva

Kurssin suorittaminen

Tapa 1: luentokurssi

- Ilmoittaudu kurssin harjoitusryhmään
- Seuraa luentoja
- Tee viikoittaiset harjoitukset
 - osallistu tavallisiin laskareihin
- Osallistu opintopiiriin ja tee ryhmätyöt
- Osallistu kurssikuulusteluihin
- Kerää riittävästi pisteitä harjoituksista, ryhmätöistä ja kokeesta

Kurssin suorittaminen

Tapa 2: erilliskuulustelu

- Ilmoittaudu ja osallistu erilliskokeeseen
- Erilliskokeen koealue määritelty kurssikuvauksessa
- Koe max 60 pistettä

Kurssin läpäisyyn tarvitset vähintään 30 pistettä

Ryhmätyöt (opintopiirit)

- Kolmeen ensimmäiseen osioon liittyy ryhmätyö
- Näistä saa pisteitä 1-4
- Ryhmissä on 2-4 opiskelijaa
- Ryhmät sovitaan harjoituksissa, niistä voi ilmoittaa myös sähköpostitse luennoijalle
- Ryhmätyöt voivat olla
 - Raportin kirjoittamista (selvitä miten jokin toimii tai on toteutettu)
 - Ohjelmointia (toteuta jokin toiminnallisuus)
 - Jotain muuta (simulaattori, yms.)

WWW Informaatio

- Kurssin kotisivu
<http://www.cs.helsinki.fi/u/niklande/opetus/kj/>
- Uutisryhmä hy.opiskelu.tkl.kj
- Kertauskysymyksiä (vanhalta KJ-II kurssilta)
<https://www.cs.helsinki.fi/i/kerola/kj2/kertaus/pract.html>
- CSMoodle <http://moodle.cs.helsinki.fi>

LUENTO 1

Yleistä käyttöjärjestelmistä

Sisältö

- Käyttöjärjestelmän palvelut
- Käyttöjärjestelmän keskeisiä osa-alueita
 - Prosessien hallinta
 - Resurssien hallinta
 - Muistin hallinta
 - Tiedostojärjestelmä
 - Siirräntäjärjestelmä (I/O)
- Laitteistosta

Tietokonejärjestelmä

- Sovellus saa laitteiston käyttöönsä KJ:n avustuksella
 - CPU ja muisti
 - oheislaitteet
- KJ tarjoaa laitteiston käytössä tarvittavat 'palvelunsa' sovellukselle
 - prosessien hallinta
 - muistinhallinta
 - siirräntäjärjestelmä
 - tiedostojärjestelmä

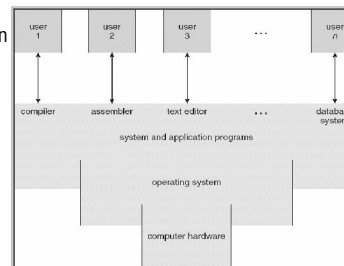


Fig 1.1 [SGG07] –

Silberschatz, Galvin and Gagne: Operating Systems Concepts with Java, Wiley, 2007

Kerrosmalli

- Peruskäyttäjä näkee vain sovellukset
 - Niiden alla on varusohjelmia ja käyttöjärjestelmä
- Sovellusohjelmoija näkee ja käyttää palvelurajapintaa (esim. funktiokirjasto)

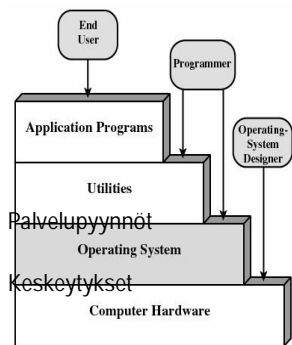


Figure 2.1 Layers and Views of a Computer System

Käyttöjärjestelmä on ...

- Valvoja (control program), joka
 - huolehtii sovellusten suorituttamisesta CPU:lla
 - antaa laitteiston sovelluksen käyttöön
 - huolehtii, että sovellus ei yksin valloita koko laitteistoa
 - tarjoaa turvallisen suoritusympäristön
- Resurssien jakaja (resource allocator), joka
 - Hallinnoi kaikkia järjestelmän resursseja, kuten muisti, oheislaitteet, jne
 - Ratkoo ristiriidat ja konfliktit mahdollisimman tehokkaasti ja reilusti
- KJ myös optimoi laitteiston käyttöä hyvän suorituskyvyn saamiseksi

KJ:n tarjoamia palveluja (1)

- Käyttöliittymä (user interface)
 - Ikkunointiympäristö, komentotulkki (shell)
- Ohjelmointiympäristö (utility programs)
 - editorit, kääntäjät, linkittäjät, debuggerit, ...
- Ohjelmien suorittaminen
 - prosessien käynnistäminen
 - kirjanpito, lataaminen muistiin ...
 - prosessien vuorottaminen
 - CPU:n käyttövuorot, resurssien varaus ja käyttö
 - prosessien tappaminen
 - resurssien vapauttaminen

KJ:n tarjoamia palveluja (2)

- Muistinhallinta
 - varaa muistitilaa sovellukselle
- Siirräntä
 - sovellus pyytää palvelua KJ:ltä
 - KJ 'komentaa' I/O-ohjaimia töihin
- Tiedostojen hallinta
 - kirjanpito tdstoista, hakemistot
 - tdstosta lukeminen / tallettaminen
- Suojaus
 - muistin suojaus ja tdstojen käyttöoikeudet
 - resurssien varaamisen kilpailutilanteet ja ristiriidat

KJ:n tarjoamia palveluja (3)

- Virhetilanteiden hallinta
 - laitteistovirheet
 - ohjelmistovirheet
 - resurssipula
- Virheistä toipuminen
 - palauttaa status tietoa sovellukselle
 - uudelleenyhteykset
 - prosessin tappaminen

Välikysymys: Mitä yleistermi resurssi tarkoittaa?

KJ:n tarjoamia palveluja (4)

- Kirjanpito / tilinpito (accounting)
 - tilastointi resurssien käytöstä
 - suorituskyvyn seuranta (esim. vastausaika)
 - järjestelmäparametrien optimointi hyvän suorituskyvyn saamiseksi
 - koneen käyttäjien laskuttaminen

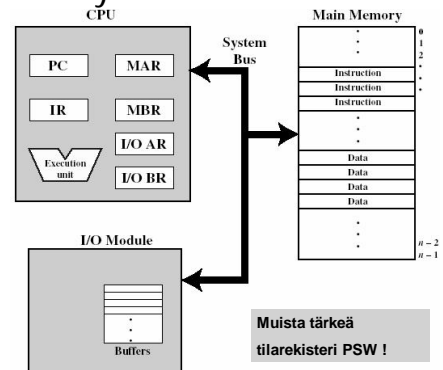
Kirjanpito voi antaa vihjeitä siitä, kuinka KJ:tä voitaisiin kehittää edelleen

Laitteistosta

- Keskusyksikkö, Käskeykli
- Keskeytys ja sen käsittely, Siirräntä
- Muistihierarkia
- Stallings, Luku 1
- = Tietokoneen Toiminta kurssin kertausta

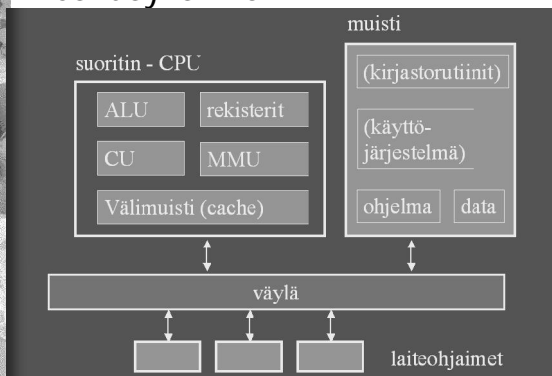
Keskusyksikkö

Fig 1.1 [Stal05]



Keskusyksikkö

ks. Tietokoneen Toiminta -kurssi



Keskusyksikön osat

- Proessori (CPU)
 - käskyjen suoritus
- (Keskus)muisti
 - tilaa ohjelmille ja niiden datalle (myös KJ on ohjelma!)
- I/O ohjaimet (I/O kanavat, I/O prosessorit...)
 - laitteistoa, joka siirtää tietoa muistiin ja oheislaitteiden välillä
 - syöttö- ja tulostuslaitteet
 - tallennuslaitteet
 - tietoliikennelaitteet
 - kommunikointi rekistereitä (I/O portteja) käyttäen
 - voivat sisältää omaa muistia (puskureita)
- Väylät
 - tiedonsiirto eri osien välillä

Muistinhallintayksikkö MMU

Memory Management Unit

- Muuttaa ohjelman sisäisen osoitteen fyysiseksi muistiosoitteeksi
 - ohjelma käyttää siirtymä alkunsa suhteen
 - laitteisto käyttää fyysisiä osoitteita
- MAR, Memory Address Register
 - muistiosoitetta varten
- MBR, Memory Buffer Register
 - Muistista noudettavan muistipaikan sisältö
 - Muistiin kirjoitettava arvo

Muistinhallintayksikkö MMU

- Kanta- ja rajarekisteriä käyttävä järjestelmä:
 - Base Register: ohjelman fyysinen alkuosoite
 - Limit Register: viimeinen kelvollinen fyysinen osoite tai viitattavissa olevan alueen pituus
- Virtuaalimuistijärjestelmä (esim.):
 - PTR, Page Table Register
 - Suoritettavan prosessin sivutaulun fyys. alkuosoite
 - TLB, Translation Lookaside Buffer
 - Viimeisimmässä osoitemuunnoksissa käytettyjä tietoja

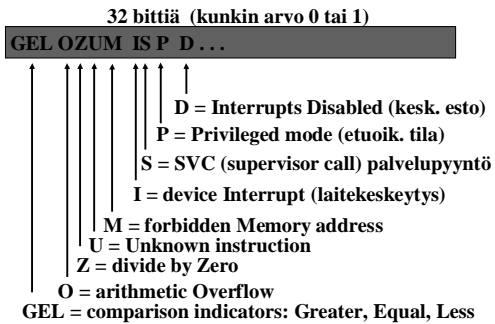
Proessorin rekistereitä

- Ohjaus- ja tilarekisterit (Control & Status Registers)
 - osa vain CPU:n sisäiseen käyttöön
 - osa vain KJ:n käyttöön
 - osa epäsuorasti ohjelmien viitattavissa
 - hyppykäsky muuttaa PC:n arvoa
 - vertailu asettaa tilarekisterin
 - ehdollinen hyppykäsky tutkii tilarekisteriä
- Yleiskäyttöiset rekisterit (User-visible registers)
 - ohjelmien käytettävissä (myös KJ:n!)
 - viitattavissa nimellä konekielen tasolla
 - datan ja osoitteiden tallettamiseen

Ohjaus- ja tilarekistereitä

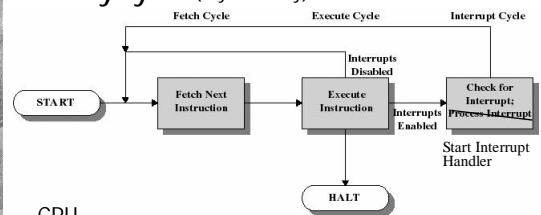
- Käskyosoitin PC (Program Counter)
 - seuraavaksi suoritettavan käskyn virtuaaliosoite
- Käskyrekisteri IR (Instruction Register)
 - suoritettavaksi noudettu käsky
- Tilarekisteri PSW (Program Status Word)
 - Tietoa laitteiston tilasta ja toiminnan siihen aiheuttamista muutoksista
- lipukkeet (flags)
 - vertailujen tuloksille, virhetilanteille
 - keskeytyksille, keskeytyksen esto / sallinen
 - etuoikeutettu tila / käyttäjällä

Esimerkki Tilarekisteristä



Käskysykli (täydennetty)

Kuva 1.7



CPU

- noutaa käskyn PC:n osoittamasta paikasta ja kasvattaa PC:n arvoa
- noutaa operandit rekistereistä tai muistista
- suorittaa käskyn yleensä ALU:ssa
- tallettaa tuloksen rekisteriin tai muistiin
- Kun keskeytys sallittu, CPU tutkii tilarekisterin ennen seuraavaa käskyä
- Jos keskeytys, suorita keskeytyksikäsitteily käskyt

Keskeytys

- Sallii CPU:n ja ohjaimien yhtäaikaisen toiminnan
- CPU vain käynnistää siirännän, ja voi jatkaa muiden käskyjen suoritusta
- Kun siirto valmis, ohjain keskeyttää CPU:n
- CPU siirtyy suorittamaan KJ:hin kuuluvaa keskeytyksikäsitteilyä
- Kun KJ käsitelty tilanteen, siirron valmistumista odottanut prosessi voi päästä taas suoritettavaksi

Moniajo (multiprogramming)

- Siirräntä hidasta, eikä sovellus voi aina jatkaa ennenkuin siirto valmis
 - esim. prosessi lukee tietoa oheislaitteelta
- ⇒ CPU odottaa toimeettomana keskeytystä

Idea: KJ ottaa suoritukseen useita prosesseja

- Kun yksi odottaa, suorita toista
- Entä, jos prosessi ei tee siirräntää?
 - Ei palvelupyynnöitä, ei keskeytyksiä
 - Muut prosessit nälkiintyvät
- Tarvitaan kello, joka keskeyttää

Keskeytykset

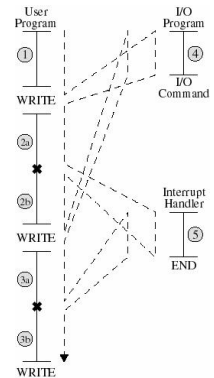
Table 1.1 Classes of Interrupts

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
- virhe - palvelupyyntö	
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
Hardware failure	Generated by a failure, such as power failure or memory parity error.

Siirräntä ja keskeytykset

Fig 1.5b [Stal05]

- Palvelupyyntö siirtää kontrollin KJ:n laiteajurille
- Ajuri alustaa ohjaimen ja antaa siirtokäskyn (4)
- Kontrolli takaisin sovellukseen
- CPU voi suorittaa sovelluksen käskyjä samaan aikaan siirron kanssa (2a)
- Kun siirto valmis, ohjain keskeyttää (x)
- Kontrolli keskeytyksäsittelijälle, josta edelleen ajurille
- Ajuri tarkastaa miten siirrossa kävi, ja tekee tarvittavat lopputoimet (5)
- Kun keskeytys käsitelty, sovelluksen suoritus jatkuu (2b)



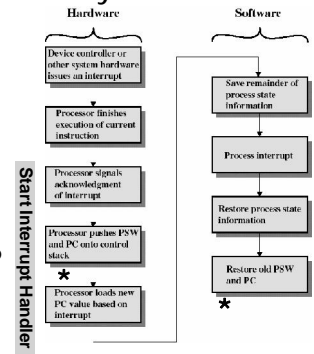
Keskeytyksäsittelijä (Interrupt handler)

- KJ:n koodia, jonka laitteisto käynnistää keskeytyksen sattuessa
- Selvittää keskeytyksen syyn
- Käynnistää toimet tilanteen hoitamiseksi
 - Siirtyminen sopivaan käsittelyrutiiniin
- Keskeytynyttä prosessia voitava jatkaa myöhemmin siitä mihin se jäi keskeytyksen sattuessa
 - CPU:n rekistereiden arvot talletettava muistiin
 - PC, PSW, muut ohjelman käyttämät rekisterit

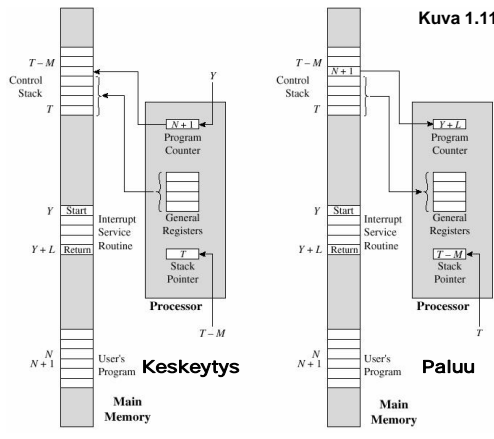
Keskeytyksäsittely (kuva 1.10)

Kirja s. 21-25

- * Etuoikeutettu tila vs. käyttäjätila
- * Keskeytysten esto vs. salliminen

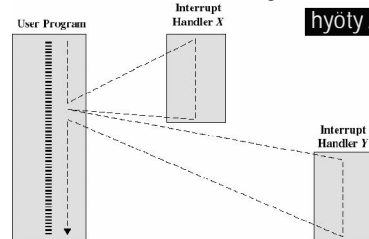


Kuva 1.11



Peräkkäinen käsittely

hyöty / haitta?



- Keskeytykset estetty käsittelyn aikana
 - PSW:n 'keskeytykset estetty' lipuke
- Uudet keskeytykset jäävät odottamaan
- Kun käsitelty, CPU tutkii taas keskeytyslipuketta

Priorisoitu käsittely

- Alemman prioriteetin keskeytyskäsittely jää kesken, jos tulee kiireellisempi keskeytys
 - ⇒ Myös keskeytyskäsittely voi keskeytyä!
- Esim. Tietoliikenneohjaimen keskeytys käsitellään välittömästi, jotta saadaan uutta tilaa saapuvale datalle

Muistihierarkia

Tan01 1.14

Typical access time	Typical capacity
1 nsec	<1 KB
2 nsec	1 MB
10 nsec	64-512 MB
10 msec	5-50 GB
100 sec	20-100 GB

Pentium 4 cache:
8 KB dataalle, 12 KB koodille, ulkoinen 256 KB

nano = 10^{-9} , mikro = 10^{-6} , milli = 10^{-3}

Välimuisti (cache memory)

- Pieni, nopea muisti / rekisterijoukko CPU:n ja keskusmuistin välissä
- CPU:n osana ja / tai ulkopuolella
- Laitetasolla, ei näy ohjelmissa / KJ:ssä
- CPU tutkii ensin löytyykö viitatus fyys. muistipaikan sisältö välimuistista
- Jos ei löydy, CPU tuo välimuistiin lohkon, joka sisältää myös viitatus muistipaikan

Osumatodennäköisyys (Hit Ratio)

= Todennäköisyys, että viitattu muistipaikka on välimuistissa

- T_1 = saantiaika välimuistista
- T_2 = saantiaika keskusmuistista
- $T_2 \gg T_1$
- Kun osumatodenn. lähellä arvoa 1, keskim. saantiaika lähellä T_1 :tä

Paikallisuus (locality of references)

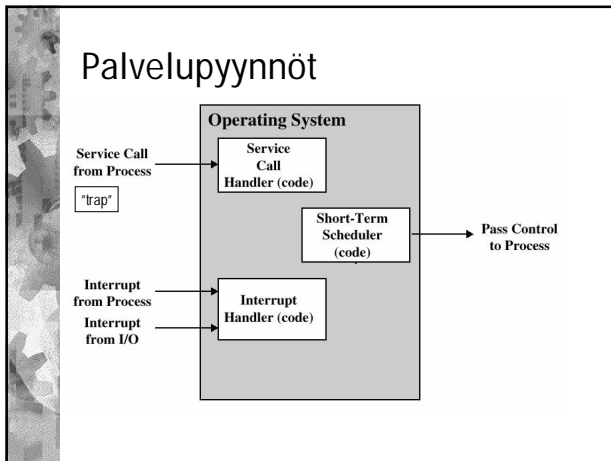
Ajallinen ja alueellinen paikallisuus:

- Esim. silmukassa suoritetaan toistuvasti samaa käskyjoukkoa
- Tiettyssä osassa koodia käytetään tyypillisesti vain tiettyjä muuttujia (data)
- ⇒ Kun ohjelma viittaa tiettyyn muistipaikkaan (käsky tai data), on tn., että se viittaa pian samaan paikkaan uudelleen tai sen lähellä oleviin muistipaikkoihin
- Osumatodenn. helposti lähellä arvoa 1 jo pienellä välimuistilla

PALVELUPYYNNÖT

Figure 2.1 Layers and Views of a Computer System

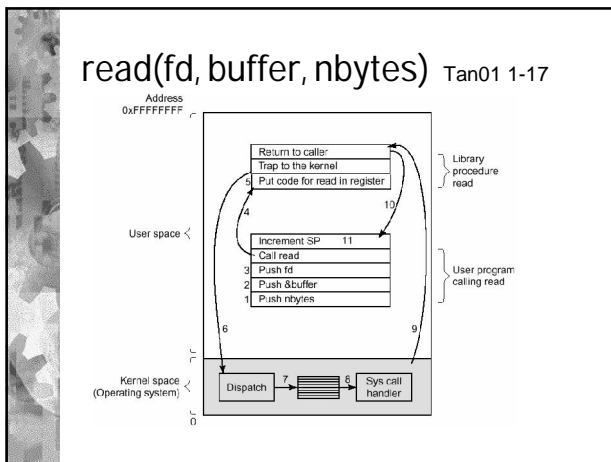
Palvelupyynnöt



Palvelupyynnöt

- Sovellus pyytää KJ:n palvelua käskykantaan kuuluvan käskyn välityksellä
 - Prosessit ja niiden välinen kommunikointi
 - Muisti
 - Tiedostot ja tiedostojärjestelmä
 - Siirräntä
 - Ym.
- Aliohjelmakutsun mekanismi
- Parametrit pinnoon, sitten käsky esim. SVC
- Palvelupyynnö aiheuttaa keskeytyksen
 - CPU etuoikeutettuun tilaan
 - CPU suorittamaan KJ:tä

read(fd, buffer, nbytes) Tan01 1-17



POSIX palvelupyynnöjä

Process management	
Call	Description
pid = fork()	Create a child process identical to the parent
pid = waitpid(pid, &status, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

File management	
Call	Description
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

Portable Operating System ("UNIX Style")

POSIX palvelupyynnöjä Tan01 1-18

Directory and file system management	
Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Miscellaneous	
Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Käyttöesimerkki

Tan01 1-19

```

#define TRUE 1

while (TRUE) {
    type_prompt( ); /* repeat forever */
    read_command(command, parameters); /* display prompt on screen */
    /* read input from terminal */

    if (fork() != 0) { /* fork off child process */
        /* Parent code. */
        waitpid(-1, &status, 0); /* wait for child to exit */
    } else {
        /* Child code. */
        execve(command, parameters, 0); /* execute command */
    }
}
    
```

"Riisuttu" komentotulkki

WIN32 API palvelupyynnöt

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time