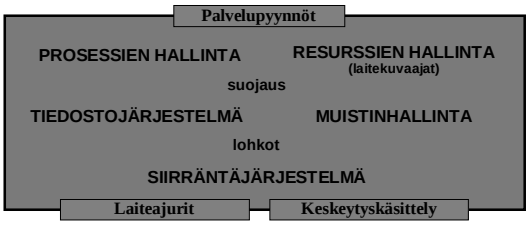
LUENTO 2

Käyttöjärjestelmän rakenne

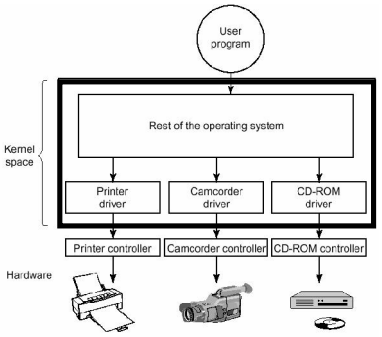
Käyttöjärjestelmän rajapinnat
Käyttöjärjestelmien kehittyminen
Nykyaikaisen käyttöjärjestelmän piirteitä

Käyttöjärjestelmän rajapinnat

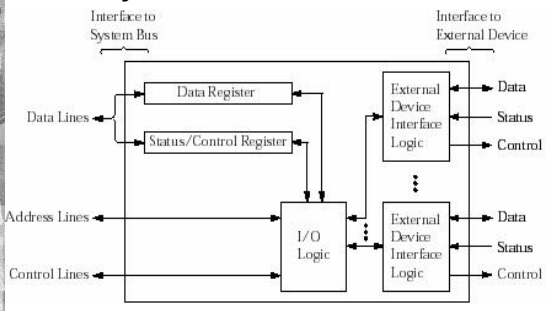


Siirräntä: Laiteajurit ja keskeytys

Fig. 5-11 [Tan01]



I/O-ohjain



I/O-ohjain

- Ohjain puskuroi väylältä tulevan / väylälle menevän datan datarekistereihinsä
  - sisäisen ja ulkoisen väylän nopeusero
- Status- ja ohjausrekisteri(t)
  - statustietoa ohjaimen / siirron tilasta
  - siirtokäskyt, osoitteet (lähde/kohde), tavumäärä
- Väylän varaus ja CPU:n keskeytys ohjausväylää käyttäen
- Liittymä laitteeseen vaihtelee tarpeen mukaan

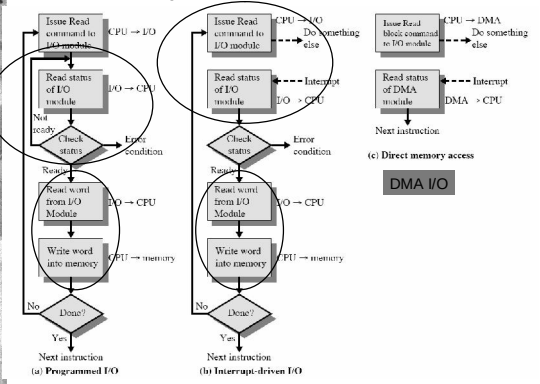
Siirtomenetelmät

Kolme perusmenetelmää

- Suora I/O (Programmed I/O)
  - ei keskeytyksiä
- Epäsuora, keskeyttävä I/O (Interrupt-driven I/O),
  - ohjain keskeyttää
- DMA-siirto (Direct Memory Access)
  - ohjain keskeyttää
  - ohjain siirtää suoraan keskusmuistiin
  - 'Älykäs' laiteohjain

Kolme eri tapa lukea lohko

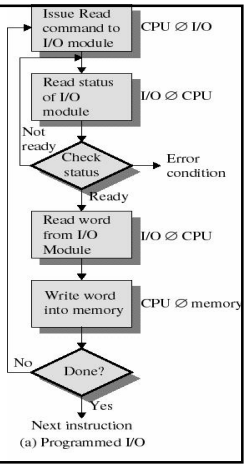
Fig 1.19 [Stal05]



suora I/O keskeyttävä I/O

Suora I/O

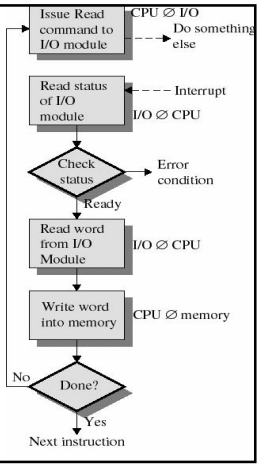
- Ei keskeytystä, CPU tutkii toistuvasti statusrekisteriä *(busy waiting, pollaus)*
- Kun siirto valmis, CPU kopioi sanan ohjaimen datarekisteristä muistiin
- CPU varattuna siirron ajan
- Vain yksinkertaisissa laitteistoissa
- Sykli toistettava kunnes kaikki siirretty
  - Ajurilla iso rooli



Epäsuora I/O

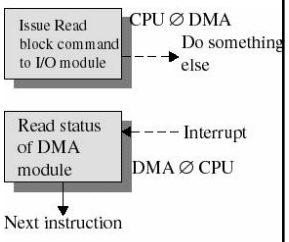
- Ohjain keskeyttää, kun siirrettävä sana datarekisterissä
- CPU siirtää datan muistiin
- CPU:n ei tarvitse pollata, kun ohjain siirtää laitteelta
- Sykli toistettava erikseen jokaiselle sanalle

è Siirto kuormittaa edelleen CPU:ta



DMA-siirto

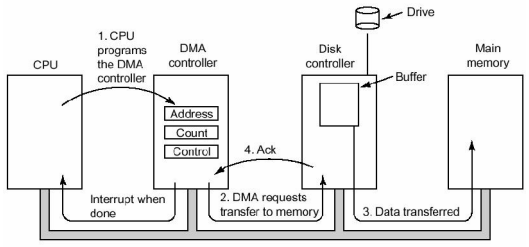
- CPU ohjaimelle: *mistä, minne, paljonko, suunta* (R / W)
- Ohjain siirtää laitteen ja muistiin välillä
- Keskeytys vasta, kun koko data siirretty
- CPU:ta vain alussa siirron käynnistykseen ja lopussa statuksen tutkimiseen



è CPU voi suorittaa siirron aikana muita prosesseja

DMA-siirto

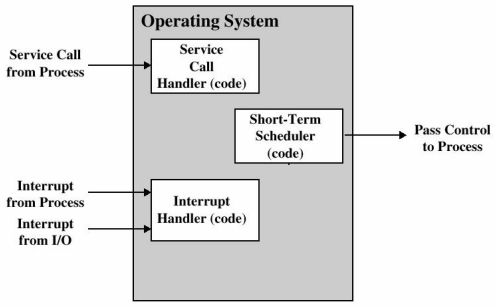
Fig. 5-4 [Tan01]



Huom: data ei kulje suorittimen rekistereiden kautta!

PALVELUPYYNNÖT

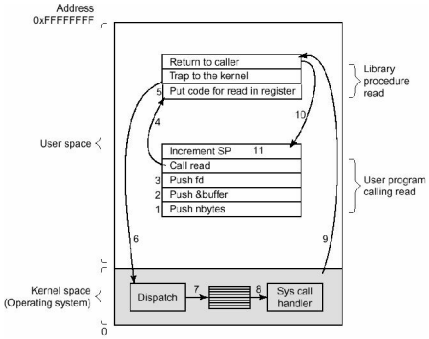
Palvelupyynnöt



Palvelupyynnöt

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

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



POSIX palvelupyyntöjä

Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, 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 palvelupyyntöjä 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) {                              /* repeat forever */
    type_prompt( );                         /* display prompt on screen */
    read_command(command, parameters);      /* 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 palvelupyyntöjä

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

Käyttöjärjestelmien kehittyminen, kehittäminen ja ylläpito

KJ:n kehittäminen ja ylläpito

- Laitteistot muuttuvat / uusia kehitetään
  - kytkimet, kortit, nauhat, levyt
  - merkkipohjaiset / graafiset päätteet
  - tuki virtuaalimuistille
  - muistin määrä kasvanut, väylät parantuneet, moniprosessorijärjestelmät, jne.
- Tietojenkäsittelytavat muuttuvat
  - interaktiiviset reaaliaikaiset järjestelmät
  - ikkunointiympäristöt
  - paikallisverkot ja Internet
  - kuvankäsittely
  - . . .

KJ:n kehittäminen ja ylläpito

- Jatkuvan kehitystarpeen vuoksi
  - modulaarinen rakenne
  - selkeät liittymät eri osien välillä
  - mahd. oliopohjainen toteutus
    - private vs. public data
- Myös KJ:ssä puutteita ja virheitä
  - paikkopaketit (patches, service packages)
  - uudet KJ-versiot
- Milloin aika tehdä KJ uudelleen alusta?

KJ:n historia lyhyesti

Perusmallit:

- Eräajojärjestelmä, yksiajo (Batch System)
- Eräajojärjestelmä, moniajo (Multiprogramming, multitasking)
- Osituskäyttöjärjestelmä (Time-Sharing)

Nykyaikaistetut mallit:

- Moniprosessorijärjestelmä (Multiprocessor)
- Verkkokäyttöjärjestelmä (Networked systems)
- Hajautettu järjestelmä (Distributed system)
- Asiakas-palvelija malli (Client-Server)

KAIKKI TARJOAVAT SAMAT PERUSPALVELUT

Eräajo
Yksiajojärjestelmä

Eräajo & yksiajo

- Ensimmäiset KJ:t 50-luvun puolivälissä
- Koneen muistissa yksink. monitoriohjelma
- Käyttäjä määritteli työnsä reikäkorteilla tai nauhalla (ns. kortinkuvat)
  erätyö = ohjauskortit + ohjelma + data
- Operaattori työnsi kortit lukijaan ja käänsi vipua
- Ohjauskortit kertoivat milloin monitorin piti ladata muita palveluohjelmia (esim. kääntäjä)
- Vain yksi työ kerrallaan suoritettavana, uusi työ ajoon vasta kun edellinen valmis

Työnohjauskieli (Job Control Language, JCL)

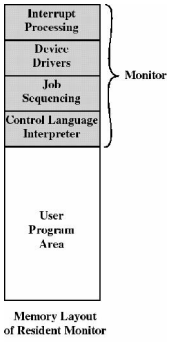
- Monitorille tarkoitettuja kortinkuvia
  - mikä ohjelma käynnistettiin
  - mitä tdstoja se käytti
  - minne tulosteet ohjattiin
- Esimerkkejä:
  - `$JOB` uuden työn alkukortti
  - `$FTN` lataa Fortran-kääntäjä ja anna kontrolli sille
  - `$LOAD` lataa käännös muistiin
  - `$RUN` aja juuri ladattu ohjelma

```
$JOB parametrit
$FTN
Ohjelmakortit
...
$LOAD
$RUN
Datakortit
...
$END
...
Seuraava erätyö
...
```

Monitori

- Jatkuvasti muistissa
- Luki kortinkuvan kerrallaan erätyötä suoritettavakseen
- Kun sovellus ladattu muistiin, suoritus hyppäsi sen alkuun
- Sovellusta suoritettiin kunnes
  - valmistui tai virhe
  - aika loppui
- Kontrolli jälleen monitorille
- Monitori luki seuraavan kortinkuvan

Kuva 2.3



Monitori ja siirräntä

- Monitori huolehti siirrännästä
  - siirrännän yksityiskohdat ei sovelluksen murheena
- I/O-käsky oli itseasiassa aliohjelmakutsu monitorin alueella olevaan koodiin
  - oma käsky, 'palvelupyyntö'
- Monitorin tarjoama palvelu
  - tarkasti, että sovellus ei vahingossa lukenut ohjauskorttia datakseen (-> liian vähän dataa?)
  - ohitti tarvittaessa kortteja, kunnes taas järkevä ohjauskortti (-> liikaa dataa?)

Monitori ja laitteistopiirteitä (1)

- Muistinsuojaus
  - Monitori suojattava sovellukselta
  - CPU:n tarkistettava muistiosoitteet
    - laitteistossa kantarekisteri BASE
- Keskeytysmekanismi
  - hallittu kontrollin siirto monitorin ja sovelluksen välillä
  - bitti PSW:ssä, keskeytyskäsittelyn alku laitetoiminto
- Kellokeskeytys
  - ettei yksi sovellus valloittanut koko laitteistoa
  - viimeistään kello aiheutti keskeytyksen
    - kontrolli taas monitorille

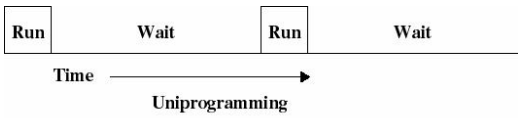
Monitori ja laitteistopiirteitä (2)

- Etuoikeutetut käskyt **(Priviledged Instructions)**
  - siirräntäkäskyt
  - muistin rajarekisterin asettaminen
  - keskeytysten esto ja salliminen
  - jos sovellus yrittää käyttää näitä käskyjä, tuloksena poikkeus 'tuntematon käskykoodi'
- Etuoikeutettu vs. käyttäjätila **(Supervisor/User mode)**
  - vain laitteisto ja monitori voi asettaa (bitti PSW:ssä)
  - CPU suorittaa etuoikeutetun käskyn vain, jos on etuoikeutetussa tilassa

Yksiajojärjestelmän heikkous

- Siirräntä erittäin hidasta
  verrattuna CPU:n nopeuteen
- CPU odottelee usein siirron valmistumista ennenkuin voi jatkaa sovelluksessa eteenpäin

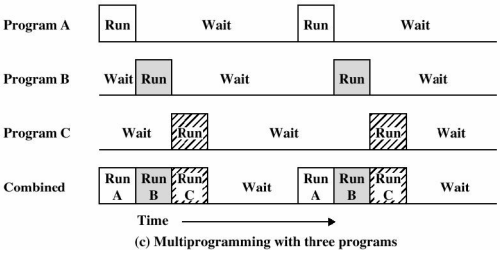
è Huono CPU:n käyttöaste



Moniajojärjestelmä

Moniajojärjestelmä

- Suoritettavaksi useita sovelluksia
  - kun yksi odottaa esim. siirrännän valmistumista, CPU suorittaa toista



Lisää laitteistovaatimuksia

- I/O-ohjain keskeyttää, kun siirräntä valmis
  - CPU voi suorittaa muuta siirron aikana
- MMU: suojaus ja ajonaik. osoitemuunnos
  - muistissa yhtäaikaa useita sovelluksia ja sovelluksen sijainti vaihtelee eri suor.kerroilla
  - Jos ei virtuaalimuistia
    - rajarekisteri LIMIT, kantarekisteri BASE
  - Jos virtuaalimuisti
    - sivutaulurekisteri PTR
    - osoitemuunnospuskuri TLB
    - sivunpuutoskeskeytys (page fault)

Lisävaatimuksia KJ:lle

- Prosessien hallinta
  - kirjanpitoa prosesseista = PCB:t
- Vuorottaminen
  - CPU toiselle prosessille, jos yksi jää odottamaan
  - tapahtumaohjattu tai aikaviipaletekniikka
  - prosessin tila: READY vs. BLOCKED
- Muistinhallinta
  - sovelluksille löydettävä tilaa muistista
  - kirjanpito vapaista ja varatuista alueista

Yksiajon ja moniajon vertailu

	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min	15 min	10 min
Memory required	50 K	100 K	80 K
Need disk?	No	No	Yes
Need terminal?	No	Yes	No
Need printer?	No	No	Yes

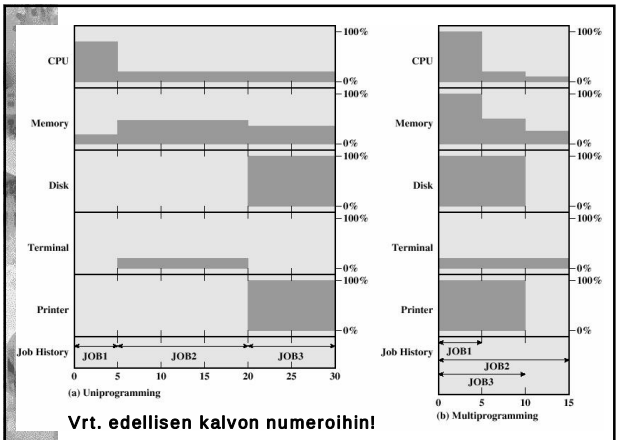
- Muistia 256 K
- Tässä ei kilpailua oheislaitteistosta

Taulukko 2.1.

Yksiajon ja moniajon vertailu

	Uniprogramming	Multiprogramming
Processor use	22%	43%
Memory use	30%	67%
Disk use	33%	67%
Printer use	33%	67%
Elapsed time	30 min	15 min
Throughput rate	6 jobs/hr	12 jobs/hr
Mean response time	18 min	10 min

Taulukko 2.2.



Vrt. edellisen kalvon numeroihin!

Osituskäyttö

- **Osituskäyttö huomioi muuttuneet käyttötavat:** interaktiivinen päätetyöskentely
  - käyttäjä voi käynnistää sovelluksen itse
  - syötteet näppäimistöltä
  - tulostus näytölle
- Ihminen koneeseen verrattuna hidas
  - tyypillinen käyttäjä tarvitsee CPU-aikaa vain 2s/min
  - järjestelmässä voi olla esim. 30 yhtäaikaista käyttäjää, eikä yksi edes huomaa muiden läsnäoloa

Osituskäyttö

- Aikaviipalekello
  - vuorottelu ei pelkästään siirrännän odottelun perusteella
  - kullekin vuorotellen aikaviipale (esim. 50-100 ms), jotta voidaan taata kaikille siedettävät vasteajat
- Prioriteetit
  - osituskäytölle suurempi prioriteetti kuin erätöille tai taustalla ajettaviin töille
    - ettei käyttäjä hermostuisi päätteensä ääressä...

NYKYAIKAISEN KJ:N PIIRTEITÄ

Uutuuksia

- Laitteistokehitys
  - moniprosessorijärjestelmät
  - nopeat verkot
  - nopeammat prosessorit
  - suurempi muisti, uudet talletusmediat
- Ohjelmistojen / käyttötapojen muutos
  - Asiakas/palvelija -malli
  - Internet ja WWW
  - Multimedia

Mikrokernel

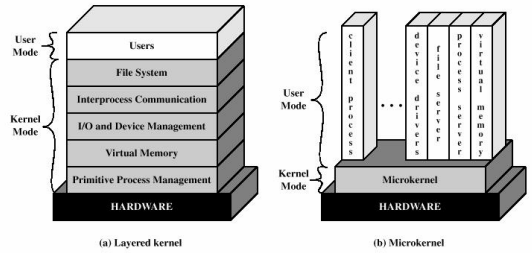
- Vain välttämättömät laitetoiminnot ytimeen, joka suoritetaan etuoikeutetussa tilassa
- keskeytyskäsittelyn alkutoimet
  - mikä / kuka aiheutti?
- vuorottamisen laitetoiminnot
  - rekistereiden kopiointia
- muistinhallinnan laitetoiminnot
  - MMU:n asetukset, suojaus
- siirrännän laitetoiminnot
  - ohjaimien käyttö, suojaus
- prosessien välinen sanomanvälitys
  - pyyntöjen välitys, kopiointia prosessien muistialueille

Mikrokernel

- Muut KJ:n palvelut 'tavallisina' prosesseina, jotka suoritetaan käyttäjätilassa
  - laiteajurit, tiedostojärjestelmä, virtuaalimuisti…
  - odottavat vuorottamista Ready-jonossa
  - eivät pääse suoraan käsiksi laitteistoon
- Toteutus perustuu sanomanvälitykseen
  - IPC, inter process communication
- Joustavuus, laajennettavuus, siirrettävyys ...
- Vrt. Monoliittinen ydin
  - KJ:n keskeiset toiminnot yhdessä ajomoduulissa
  - yleisempää, nopeampaa

KJ:n ydin

Kuva 4.10



Monoliittinen ydin

Säikeet (multithreading)

- Prosessi voi jakautua yhteen tai useampaan säikeeseen, jotka yhtäaikaa Ready-jonossa
- *Yhden* prosessin säikeet käyttävät *yhteistä koodia, data-aluetta ja resursseja* sekä pääosaa prosessin kuvaajasta
- Säikeen *luonti ja lopettaminen nopeampaa* kuin prosessin
- Saman prosessin säikeiden *vuorottaminen nopeampaa* kuin eri prosessien vuorottaminen

Moniprosessorijärjestelmä

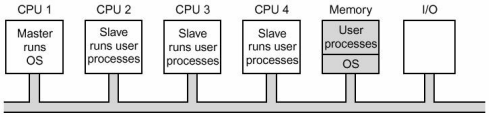


Fig. 8-8. A master-slave multiprocessor model.

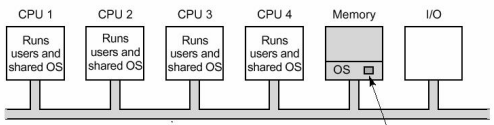
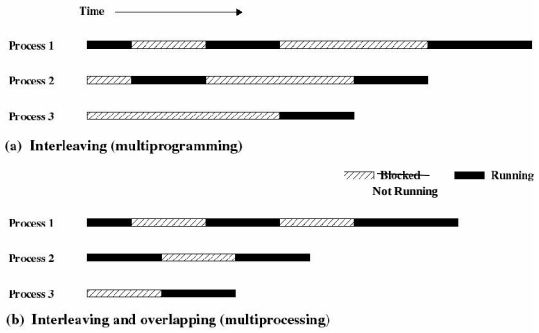


Fig. 8-9. The SMP multiprocessor model.

SMP

- Koneessa useita CPU:ita
  - kaikki rakenteeltaan ja tehtäviltään samanlaisia
    - SMP, Symmetric Multiprocessing
  - aidosti rinnakkainen suoritus
  - kukin voi suorittaa KJ:tä tai sovellusta
- Muu laitteisto yhteiskäytössä
  - muisti, väylät, I/O-laitteet
- Useamman CPU:n mukanaolo ei vaikuta normaaliin ohjelmointiin
  - KJ:ssä sensijaan paljonkin uutta mietittävää
- Tehokkuus, vikasietoisuus, laajennettavuus

3 prosessia ja SMP (2 CPU:ta)



Verkkojärjestelmä

- Useita erillisiä (mahd. erilaisia) solmukoneita
- Kullakin koneella oma KJ ja omat prosessit
- Mahd. yhteiskäytössä oleva tdstojärjestelmä
- Globaali käyttäjien tunnistus

Verkkojärjestelmä
- Käyttäjä tuntee ja käyttää koneita nimeltä
- Toisella koneella olevien tiedostojen käyttö
- KJ:t voivat olla erilaisia eri koneissa

Hajautettu järjestelmä

- Useita erillisiä koneita
- Kullakin koneella oma KJ ja omat prosessit
- Mahd. yhteiskäytössä oleva tdstojärjestelmä
- Globaali käyttäjien tunnistus

Hajautettu järjestelmä
- Käyttäjän ei tarvitse tuntea koneita nimeltä
- KJ hoitaa mm. kuormantasauksen
- Globaali KJ (kaikissa samanlainen)

Asiakas-palvelija malli

- Sovellus jaettu useampaan osaan
  - esim. WWW-palvelija ja selain (käyttöliittymä)
- Asiakas ja palvelija voivat sijaita eri koneissa
  - WWW-palvelija konehuoneen palvelimella, selainohjelma työhuoneen koneella
- tai samassa koneessa
  - ikkunamanageri ja sovellusohjelma
- Palvelija palvelee useita asiakkaita
- Sanomanvälitys
  - TCP/IP-protokolla, etäproseduurikutsu

Reaaliaikajärjestelmä

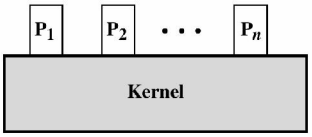
- Tarve reagoida ulkopuolisiin tapahtumiin
  - Ohjausjärjestelmät: laboratoriokokeet, teollisuus, lentoliikenne, teleliikenne, robotiikka
- Tapahtumat tulevat reaaliajassa
  - Ehdittävä käsitellä ennen uutta
- Hard Real-time vs. Soft Real-time
  - Ei saa missata aikarajoja (deadline) vs. yrittää parhaansa, saa joskus myöhästyäkin
- Periodinen vs. aperiodinen
  - Ajallinen tai määrällinen säännöllisyys
  - Alku- ja/tai päättymisajalle aikaraja

KJ:N SUORITTAMISESTA

KJ:n suorittamisesta

- Myös KJ eräs CPU:n suorittamista käskykokoelmista
- Käyttäjätilassa / etuoikeutetussa tilassa
- KJ:n osat käsittelevät yhteisiä data-alueita
  - melkein kaikki käyttävät PCB:tä
- Onko KJ myös prosessi?

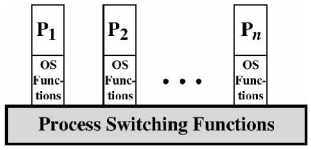
j KJ etuoikeutetussa tilassa



- Prosessi vain käyttäjätilan käsite
  - KJ:n osat eivät jonota
- KJ:llä omat muistialueensa: koodi, data, pino
- KJ:n osat suoritetaan omillaan etuoik. tilassa
  - oikeus tehdä kaikkia KJ:n toimintoja kaikissa osissa

~ vanha monoliittinen KJ

k KJ prosessin ympäristössä

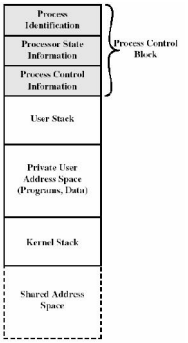


- KJ yhteiskäyttöisellä muistialueella
  - kaikkien prosessien osoiteavaruudessa
- Prosessi itse suorittaa KJ:n rutiineja
  - hallittu siirtyminen keskeytyksellä, etuoikeutettu tila
- Kontrolli prosesseilta poissa vain, kun synkronointi tai vuorottaminen vaatii

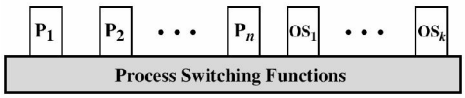
~ uudempi monoliittinen KJ

KJ prosessin ympäristössä

- KJ:n koodi ja data yhteisellä muistialueella
- Prosessi käyttää kernel-pinoa, kun suorittaa KJ:n koodia, muulloin normaalia pinoaan
- Prosessi voi odottaa KJ:n koodissa
- Useita KJ:n osia voi olla yhtäaikaa kesken eri prosessien ympäristöissä
  - suoritukseen vuorottajan kautta



l KJ = joukko palveluprosesseja



- Monet KJ:n palveluista erillisiä prosesseja
  - odottavat Blocked/Ready-jonossa
  - kullakin oma osoiteavaruus
  - tarvittaessa etuoikeutetussa tilassa, erilaisia oikeuksia
- Vuorottaminen prosessien ulkopuolella
- Sanomanvälitys: pyyntö-vastaus mekanismi
  - palvelupyyntö: lähetä / vastaanota sanoma
  - sopii myös moniprosessori / hajautettuihin järjestelmiin
- Jos ytimessä vain laiteriippuvat toiminnot
  = mikrokernel