LUENTO 4

PROSESSIT JA NIIDEN HALLINTA

Stallings, Luku 3

1

Sisältöä

- Prosessi, prosessin kuvaaja
- Prosessien hallinta
- Prosessin tilat
- KJ:n perustietorakenteita
- KJ:n suorittamisesta

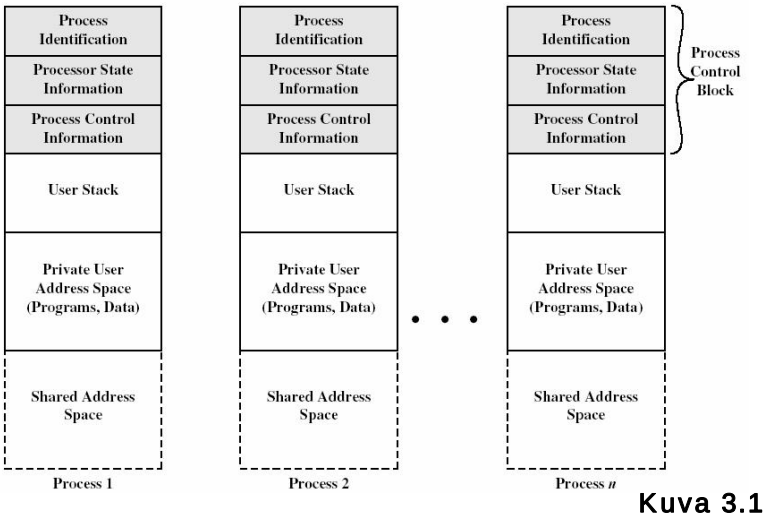
2

Prosessi

- Moniajo perustuu prosessikäsitteeseen
- Prosessi = ohjelman suoritus prosessorissa
  - koodi, data, pino, prosessin kuvaaja PCB
  - voi koostua useasta säikeestä
- Prosessit voivat suorittaa yhtäaikaa samaa ohjelmakoodia
  - vapaakäyntisyys (reentrancy)
  - Yhteinen koodialue
  - kullakin oma data-alue, pino, PCB
- Täsmällinen määrittely riippuu jossain määrin järjestelmästä ja ohjelmointikielestä

3

Prosessi virtuaalimuistissa



Kuva 3.12

4

Prosessi virtuaalimuistissa

- Kukin prosessi käyttää virtuaaliosoitteita
  - osoitteet suhteellisia prosessin alun suhteen
    - MMU tekee osoitemuunnoksen ajoaikana
  - prosessin alueiden ei tarvitse sijaita fyysisesti peräkkäin muistissa tai olla jatkuvasti muistissa
    - MMU ja KJ huolehtivat alueiden muistissaolosta
  - prosessit voivat käyttää myös yhteisiä muistialueita
- Prosessin alueet kirjattu PCB:hen
  - base ja limit (fyys. alkuosoite ja pituus) tai
  - sivutaulu (missä sivutiloissa sivut sijaitsevat) tai
  - segmenttitaulu (alkuosoitteet ja pituudet)

5

Prosessi

- Koodi = suoritettavat käskyt
- Data = muuttujat
- Pino = työtilaa
  - Aliohjelman/systeemikutsun parametrinvälitys
- Prosessin kuvaaja, PCB
  = hallinnolliset rakenteet
  - tunnistus
  - vuorottajan tarvitsemaa tietoa
    - mm. prosessorin tila (tallealue rekistereille)
  - tietoja varatusta muistista
  - tietoja avatuista tdstoista
  - ym.

6

PCB: tunnistus

- Yksikäsitteinen numero
  - pid = process identification
- Omistajan tiedot
  - käyttäjän ja ryhmän tunniste
    uid = user id, gid = group id
  - yleensä sama kuin prosessin käynnistäjällä
    - saatu kun käyttäjä ottaa istunnon koneeseen
- Mammaprosessin tunniste
  - mikä prosessi loi tämän prosessin
    - kopioitu mammaprosessin kuvaajasta

7

PCB: tallealue rekistereille

Keskeytys:

- Keskeytyskäsittelyn jälkeen
  tav. sama prosessi saa jatkaa
  - laitteisto tallettaa PC:n ja PSW:n pinoon
  - käsittelijä tallettaa käyttämänsä rekisterit pinoon
  - kun keskeytys käsitelty, palautetaan takaisin CPU:hun

Prosessin vaihto:

- CPU toiselle prosessille
  - keskeytyskäsittelyn lopuksi vuorottajaan
  - vuorottaja tallettaa rekistereiden arvot PCB:hen
  - prosessin tila saattaa vaihtua
  - päivitettävä myös aika- ja viitelaskureita

8

PCB: vuorottaminen

- Prosessin tila: Running, Ready, Blocked...
- Prioriteetti
  - oletus, maksimi, minimi
  - määrää sijainnin jonoissa
  - suuri prioriteetti ð saa useammin CPU-aikaa
  - voi vaihdella dynaamisesti
- Aikalaskureita
  - paljonko käyttänyt aikaa CPU:ssa, odotuksessa
  - voi vaikuttaa prioriteettiin
- Mitä tapahtumaa odotetaan

9

PCB: muistinhallinta

- Muistialueen alkuosoite ja pituus
  - Base ja Limit
- tai Sivu / segmenttitaulun fyys. osoite
  - taulu erillisellä muistialueella
  - prosessin vaihdossa alkuosoite MMU:hun
- Yhteiskäyttö
  - sama sivu / segmentti esiintyy eri prosessien muistivaraustauluissa
  - käyttöoikeudet: esim. R / W / RW

10

PCB: tiedostojärjestelmä

- Tiedostokuvaajataulu (file descriptor)
  - alkio per avattu tdsto
  - pääsy muihin tdstoon liittyviin rakenteisiin
    - kaikille yhteistä tietoa!
      - missä tdstoon kuuluvat lohkot
      - käyttöoikeudet
      - tdstolukot
    - kullakin oma luku/kirjoituspositio
- Työhakemiston polkunimi
  - suhteellisen tdstonimen käyttö
- Luotavien tiedostojen (oletus)käyttöoikeudet

11

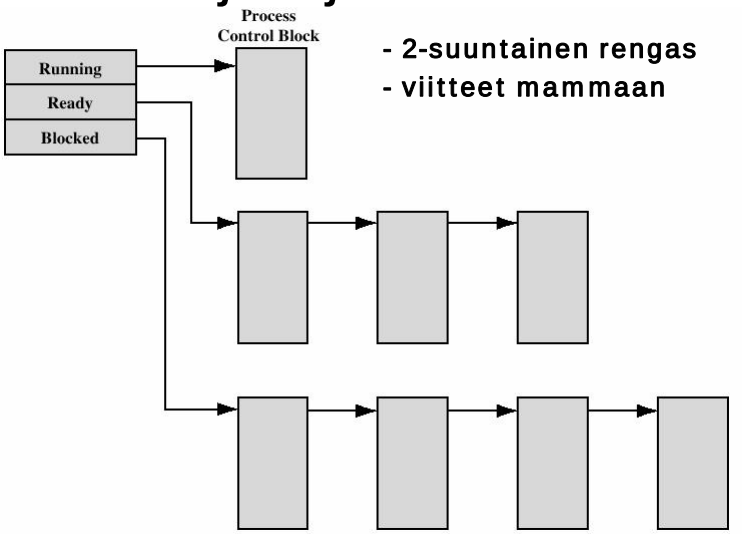
PCB: muuta

- Viitteitä muihin prosessin kuvaajiin
  - ks. kuva 3.13
  - jonossa edeltävä ja seuraava
  - viite mammaprosessin kuvaajaan
  - . . .
- Prosessien välinen kommunikointi
  - lipukkeita, semaforeja
  - käsittelyä odottavat signaalit
  - yhteiskäytössä oleva muisti
- Ym.

12

Prosessijonoja

Kuva 3.13

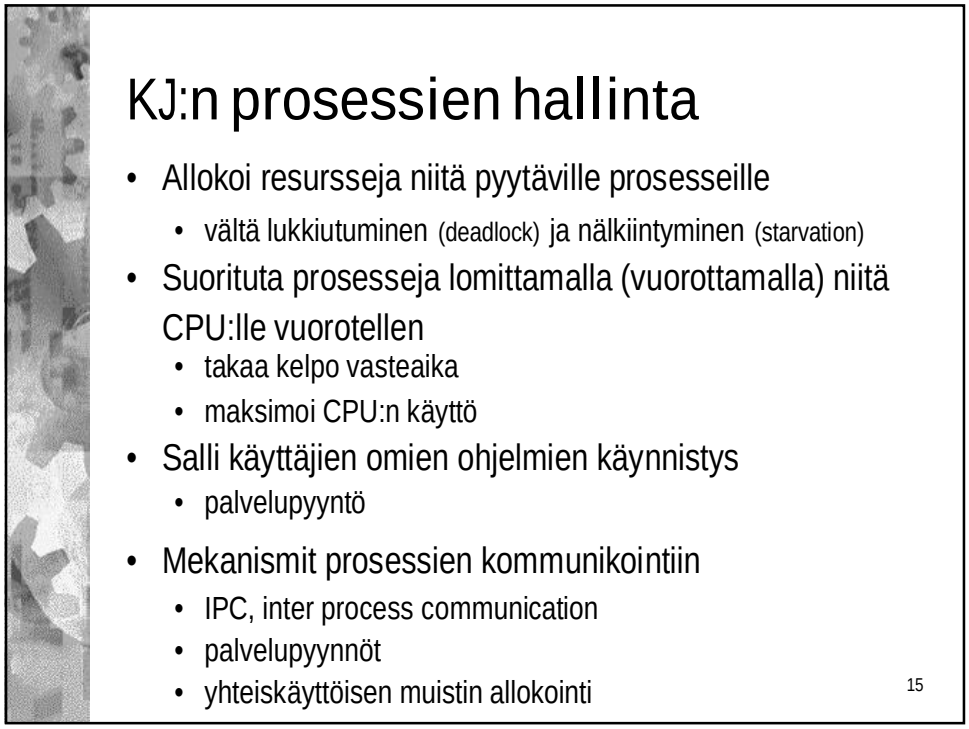


- 2-suuntainen rengas
- viitteet mammaan

13

PROSESSINHALLINTA

14



KJ:n prosessien hallinta

- Allokoi resursseja niitä pyytäville prosesseille
  - vältä lukkiutuminen (deadlock) ja nälkiintyminen (starvation)
- Suorituta prosesseja lomittamalla (vuorottamalla) niitä CPU:lle vuorotellen
  - takaa kelpo vasteaika
  - maksimoi CPU:n käyttö
- Salli käyttäjien omien ohjelmien käynnistys
  - palvelupyyntö
- Mekanismit prosessien kommunikointiin
  - IPC, inter process communication
  - palvelupyynnöt
  - yhteiskäyttöisen muistin allokointi

15

Prosessi syntyy, kun ...

- Prosessi suorittaa käskyt, joilla käynnistetään uusi prosessi
  - parametrit pinoon ja palvelupyyntökeskeytys
- Prosessi pyytää muita KJ:n palveluja

  ð KJ käynnistää prosessin palvelemaan
  - esim. tulostus kirjoittimelle ð taustaprosessi
  - palvelija keskustelee asiakkaiden kanssa
    - yksi prosessi per yhteys
- Esim: Käyttäjä avaa istunnon koneeseen

  Käyttäjä käynnistää sovelluksen

  Erätyö käynnistyy

16

Taulukko 3.1

Table 3.1 Reasons for Process Creation

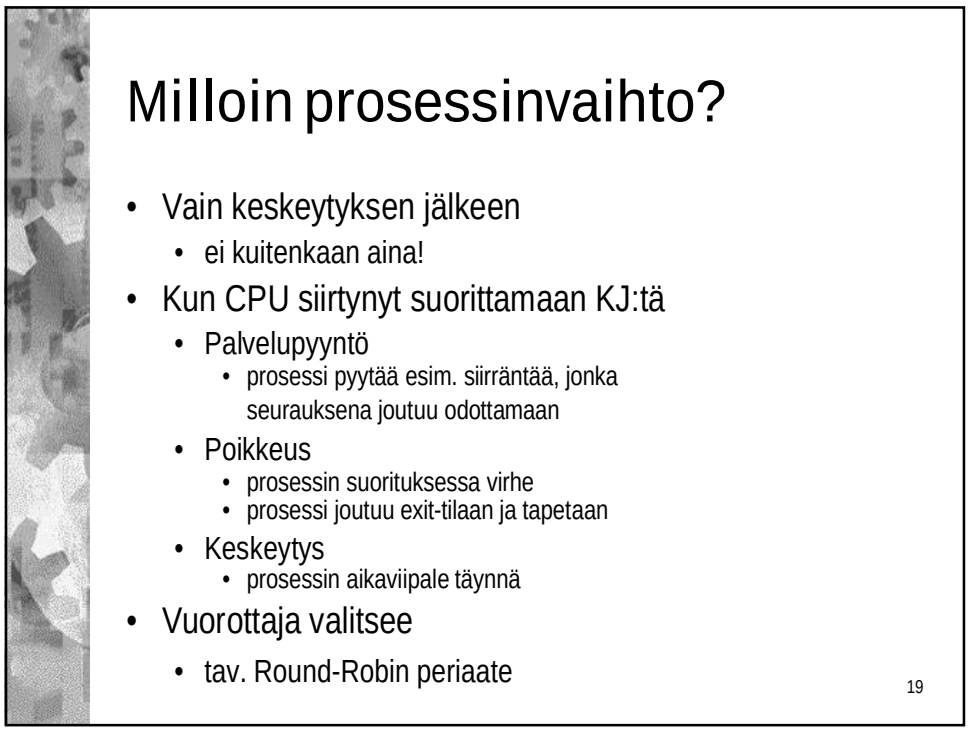
New batch job	The operating system is provided with a batch job control stream, usually on tape or disk. When the operating system is prepared to take on new work, it will read the next sequence of job control commands.
Interactive logon	A user at a terminal logs on to the system.
Created by OS to provide a service	The operating system can create a process to perform a function on behalf of a user program, without the user having to wait (e.g., a process to control printing).
Spawned by existing process	For purposes of modularity or to exploit parallelism, a user program can dictate the creation of a number of processes.

17

Prosessin luonti - toimenpiteet

- Luo prosessin kuvaaja (PCB)
  - KJ valitsee yksikäsitteisen prosessinumeron
- Varaa tilaa muistista (tarvittaessa)
  - koodi, data, pino
  - koko: oletusarvot / annetut arvot
  - alusta yhteiskäytön rakenteet
- Alusta PCB:n
  - nollaa kenttiä, aseta alkuarvoja, kopioi mammalta
  - tila=Ready (tai Ready-Suspended)
  - ei avoimia tiedostoja, ei varattuja resursseja ... tai perii mammalta
- Liitä kuvaaja muihin rakenteisiin
  - viite mammaprosessiin, liitä Ready-jonoon

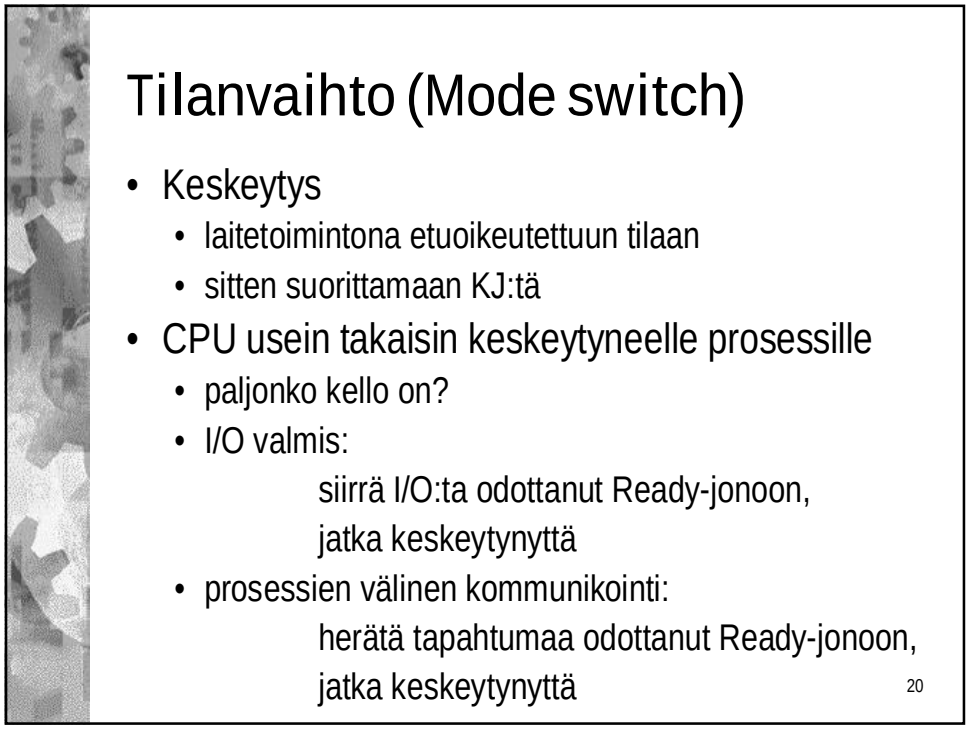
18



Milloin prosessinvaihto?

- Vain keskeytyksen jälkeen
  - ei kuitenkaan aina!
- Kun CPU siirtynyt suorittamaan KJ:tä
  - Palvelupyyntö
    - prosessi pyytää esim. siirräntää, jonka seurauksena joutuu odottamaan
  - Poikkeus
    - prosessin suorituksessa virhe
    - prosessi joutuu exit-tilaan ja tapetaan
  - Keskeytys
    - prosessin aikaviipale täynnä
- Vuorottaja valitsee
  - tav. Round-Robin periaate

19



Tilanvaihto (Mode switch)

- Keskeytys
  - laitetoimintona etuoikeutettuun tilaan
  - sitten suorittamaan KJ:tä
- CPU usein takaisin keskeytyneelle prosessille
  - paljonko kello on?
  - I/O valmis:
    siirrä I/O:ta odottanut Ready-jonoon,
    jatka keskeytynyttä
  - prosessien välinen kommunikointi:
    herätä tapahtumaa odottanut Ready-jonoon,
    jatka keskeytynyttä

20

Tilanvaihto (mode switch)

- Kaikkia rekistereitä ei tarvitse tallettaa muistiin
  - PC ja PSW aina laitetoimintona pinoon
  - keskeytyskäsittelijä tallettaa pinoon vain ne, joita käyttää koodissaan
- PCB:hen ei tarvitse koskea
  - vähän yleisrasitetta
- Paluu:
  - **kopioi rekisterit pinosta takaisin CPU:hun**
- Vuorottajaan, jos tarve vaihtaa prosessia

21

Vuorottaja (short-term scheduler)

- Valitsee seuraavaksi suoritettavan prosessin ja antaa CPU:n sille
  - edellinen Blocked-tilaan
  - aikaviipale täyttyi
- Prosessin vaihdossa CPU suorittaa vuorottajan käskyjä
- CPU-aikaa tasapuolisesti prosesseille
  - aikaviipaleet
  - tarvittaessa KJ nostaa / laskee prosessin prioriteettia
    - käyttänyt paljon CPU:ta ð prioriteetti laskee
    - odotellut paljon I/O:ta ð prioriteetti nousee
    - KJ:n prosesseilla suurin prioriteetti

22

Prosessin vaihto

- Rekisteriarvot pinosta+CPU:sta PCB:hen
- Päivitä aikalaskureita ym.
- Päivitä prosessin tila (Ready/Blocked/Exit...)
- Liitä tilan mukaiseen jonoon
- Valitse seuraava prosessi suoritettavaksi
  - Ready-jonon ensimmäinen
  - Ready ð Running
- Alusta MMU
  - ei-virtuaalimuistia: aseta Base ja Limit
  - virtuaalimuisti: nollaa TLB:n validibitit, aseta PTR
- Palauta rekistereiden arvot CPU:hun

23

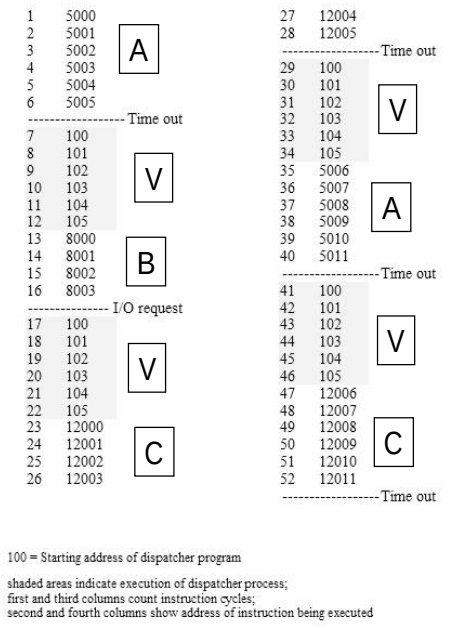


Figure 3.4 Combined Trace of Processes of Figure 3.2

Prosessit: A,B ja C
Vuorottaja (Dispatcher) V

24

Prosessien tilamuutokset aikajanalla

A
B
C
V

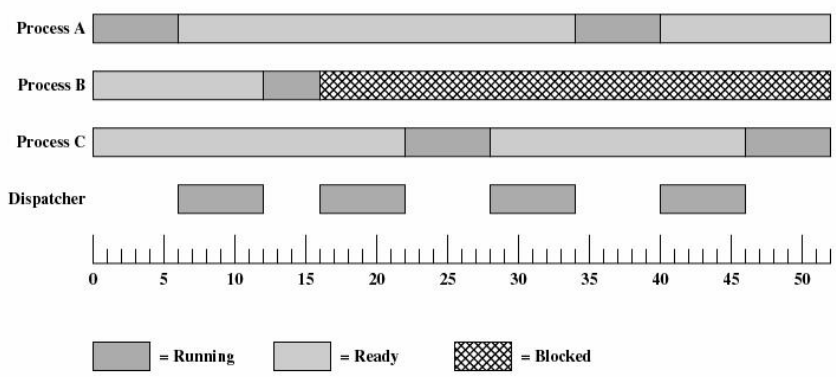


Figure 3.7 Process States for Trace of Figure 3.4

25

Prosessi päättyy, kun ...

- Prosessi itse sitä pyytää
  - KJ:n tarjoama palvelu, palvelupyyntö
  - esim. TITO-kurssilla SVC SP,=HALT
- Virhetilanne koodissa tai laitteistossa
  - poikkeukset
  - parempi lopettaa kuin nilkuttaa virhetuloksia
- Esim: Käyttäjä lopettaa istuntonsa
  Käyttäjä lopettaa sovelluksen
  Erätyö loppuu

26

Prosessi päättyy, kun ...

- KJ niin päättää
  - puutteelliset oikeudet esim. tiedoston käyttöön
  - huomattuaan lukkiutumisen
  - odotukseen liitetty ajastin laukeaa
- Mammaprosessi pyytää lapsiprosessin päättymistä
- Mammaprosessi päättyy
  - saattaa myös lapsiprosessi päättyä

27

Taulukko 3.2 (1/2)

Table 3.2 Reasons for Process Termination

Normal completion	The process executes an OS service call to indicate that it has completed running.
Time limit exceeded	The process has run longer than the specified total time limit. There are a number of possibilities for the type of time that is measured. These include total elapsed time ("wall clock time"), amount of time spent executing, and, in the case of an interactive process, the amount of time since the user last provided any input.
Memory unavailable	The process requires more memory than the system can provide.
Bounds violation	The process tries to access a memory location that it is not allowed to access.
Protection error	The process attempts to use a resource such as a file that it is not allowed to use, or it tries to use it in an improper fashion, such as writing to a read-only file.
Arithmetic error	The process tries a prohibited computation, such as division by zero, or tries to store numbers larger than the hardware can accommodate.

28

Taulukko 3.2 (2/2)

Table 3.2 Reasons for Process Termination

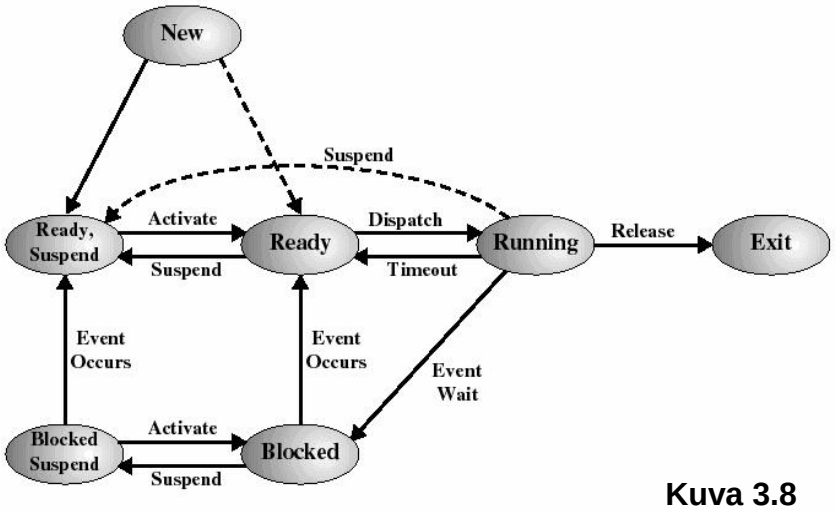
Time overrun	The process has waited longer than a specified maximum for a certain event to occur.
I/O failure	An error occurs during input or output, such as inability to find a file, failure to read or write after a specified maximum number of tries (when, for example, a defective area is encountered on a tape), or invalid operation (such as reading from the line printer).
Invalid instruction	The process attempts to execute a nonexistent instruction (often a result of branching into a data area and attempting to execute the data).
Privileged instruction	The process attempts to use an instruction reserved for the operating system.
Data misuse	A piece of data is of the wrong type or is not initialized.
Operator or OS intervention	For some reason, the operator or the operating system has terminated the process (for example, if a deadlock exists).
Parent termination	When a parent terminates, the operating system may automatically terminate all of the offspring of that parent.
Parent request	A parent process typically has the authority to terminate any of its offspring.

29

PROSESSIN TILAT

30

Prosessin tilakaavio (7 tilaa)



Kuva 3.8

31

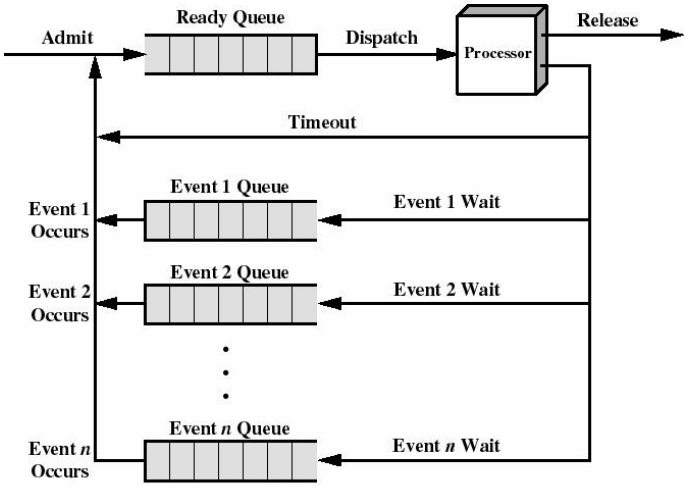
Prosessin tilat

- Ready
  - prosessi voisi edetä, jos saisi CPU:n käyttöönsä
  - odottaa Ready-jonossa esim. prioriteetin mukaan
- Running
  - prosessi käyttää parhaillaan prosessoria
  - yksi prosessi per prosessori
- Blocked
  - prosessi odottaa tapahtuman valmistumista (esim I/O, synkronointi, ajastus)
  - kullakin laitteella / tapahtumalla oma jono

32

Prosessijonot

Kuva 3.7



33

Prosessin tilat

- New
  - KJ luonut lapsiprosessin,
    - prosessille annettu tunniste
    - prosessille luotu hallinnolliset tietorakenteet
  - mutta ei kelpuuta sitä vielä suoritettavaksi
    - esim. vapaata muistia ei riittävästi
    - liian suuri moniajoaste, liikaa heittovaihtoa ...
- Exit
  - suoritus päättynyt,
    - ei kelpaa enää suoritettavaksi
  - mutta 'saattohoito' tekemättä
    - hallinnolliset rakenteet (lähinnä PCB) olemassa muita sovelluksia varten
    - esim: laskutus, tilastot

34

Prosessin tilasiirtymät

- New ð Ready
  - resursseja riittävästi käytettävänä
    - esim. prosessorin käyttöaste laskenut alle sopivan rajan
    - muistissa riittävästi vapaata tilaa
- Ready ð Running
  - vuorottaja valitsee suoritukseen Ready-jonon ensimmäisen prosessin
- Running ð Ready
  - prosessin aikaviipale täynnä
  - suuremman prioriteetin prosessi Ready-tilassa

35

Prosessin tilasiirtymät

- Running ð Blocked
  - prosessi pyytää KJ:ltä palvelua, jonka valmistumista joutuu odottamaan
    - tarvittava resurssi varattu
    - odottaa siirrännän valmistumista
    - odottaa toisen prosessin etenemistä sopivaan vaiheeseen (prosessin välinen kommunikointi)
  - sivunpuutoskeskeytys
- Blocked ð Ready
  - prosessin odotus päättyy
    - tarvittu resurssi vapautui
    - siirräntä valmistui
    - toinen prosessi saavutti synkronointikohdan

36

Prosessin tilasiirtymät

- Running ð Exit
  - prosessin suoritus päättyy
    - normaali / virhetilanne
  - KJ vapauttaa resurssit PCB:tä lukuunottamatta
  - odotettava, että joku toinen prosessi kokoaa kirjanpidolliset tiedot PCB:stä
- Mikä tahansa tila ð Exit
  - KJ tai omistaja voi tappaa
  - mammaprosessi päättyy
- Exit ð
  - kun 'saattohoito' tehty, KJ vapauttaa PCB:n

37

Heittovaihto (swapping)

- Prosessi odottaa siirron valmistumista kauan
  - paljon prosesseja Blocked-tilassa
  - KJ voi ottaa suoritettavaksi lisää prosesseja
  - riittääkö muistia?
- Jos muistitilasta puutetta, KJ voi siirtää kokonaisia prosesseja levylle
  - liian suuri moniajoaste aiheuttaa ruuhkautumista
- PCB jää aina muistiin!
- Kun tilaa jälleen riittävästi, KJ tuo takaisin
  - ennaltanouto / tarvenouto
- Myös virtuaalimuistin yhteydessä voi olla tarvetta heittovaihtoon

38

Heittovaihdon tilat

Heittovaihdon lisätilat tilakaavioon:

- Blocked Suspend
  - Blocked-prosessi heittovaihdettu muistista levylle
- Ready Suspend
  - Ready-prosessi heittovaihdettu muistista levylle

Blocked ~ estynyt
Suspend ~ erotettu määräajaksi, hyllytetty, lykätty toistaiseksi
Monissa kirjoissa tilan nimenä kuvaavampi Swapped Out

39

Heittovaihdon tilasiirtymät

- New ð Ready Suspend
  - KJ ottanut prosessin suoritettavaksi (=PCB luotu), mutta muistissa ei vielä tilaa uudelle prosessille
- Blocked ð Blocked Suspend
  - KJ tarvitsee lisätilaa Ready-prosesseille
  - KJ tarvitsee tilaa uusille prosesseille
- Blocked Suspend ð Ready Suspend
  - Tapahtuman odotus päättyy, prosessi voisi jatkaa
- Blocked Suspend ð Blocked
  - Muistissa taas tilaa, odotettavissa että odotus päättyy
  - Suuri prioriteetti

40

Heittovaihdon tilasiirtymät

- Ready ð Ready Suspend
- Running ð Ready Suspend
  - KJ haluaa lisää muistitilaa, eikä yhtään Blocked-prosessia heittovaihdettavaksi
- Ready Suspend ð Ready
  - CPU:n käyttöaste laskenut riittävän alas
  - ei prosesseja Ready-jonossa
  - muistissa jälleen reilusti tilaa (ennakointi)
    - Yl. siirtymän Blocked --> Blocked Suspend seurausta

Huomautus: Suspend-tilat eivät välttämättömiä, jos virtuaalimuisti (MMU huomaa puutoksen)

41

Hyllytyksen syitä

Table 3.3 Reasons for Process Suspension

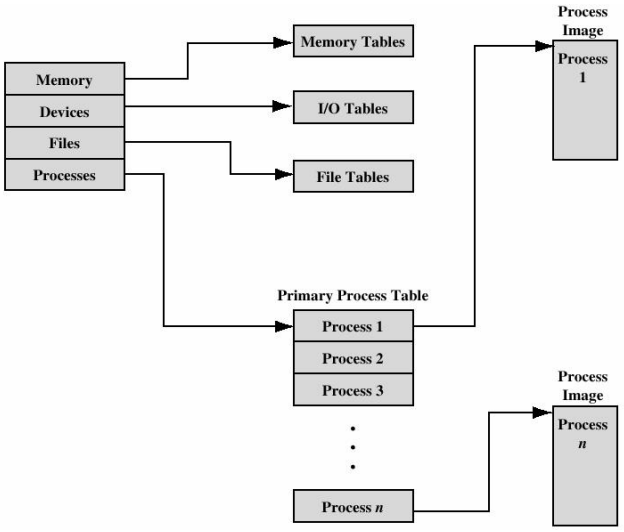
Swapping	The operating system needs to release sufficient main memory to bring in a process that is ready to execute.
Other OS reason	The operating system may suspend a background or utility process or a process that is suspected of causing a problem.
Interactive user request	A user may wish to suspend execution of a program for purposes of debugging or in connection with the use of a resource.
Timing	A process may be executed periodically (e.g., an accounting or system monitoring process) and may be suspended while waiting for the next time interval.
Parent process request	A parent process may wish to suspend execution of a descendent to examine or modify the suspended process, or to coordinate the activity of various descendents.

42

KJ:N PERUSTIETORAKENTEITA

43

KJ:n perustietorakenteita Kuva 3.10



44

KJ:n perustietorakenteita

- Prosessitaulu
  - tietoa kaikista järjestelmän prosesseista
  - kullekin oma alkio = prosessin kuvaaja **PCB**
  - alkiot linkitetty Ready ja Blocked-jonoihin

45

KJ:n perustietorakenteita

- Muistivaraustaulut / -listat
  - missä vapaata / varattua muistitilaa
    - esim. yksi globaali sivutilataulu
  - mitkä alueet kuuluvat millekin prosessille
    - prosessikohtaiset sivutaulut
    - PCB:ssä esim. sivutaulun fyysinen osoite
  - kenellä käyttöoikeuksia muistialueisiin
    - yhteiskäyttö joskus sallittua
  - virtuaalimuistin ja heittovaihdon toteutus
    - heittovaihtoalue levyllä
    - sivutaulun alkioissa läsnäolobitit

46

KJ:n perustietorakenteita

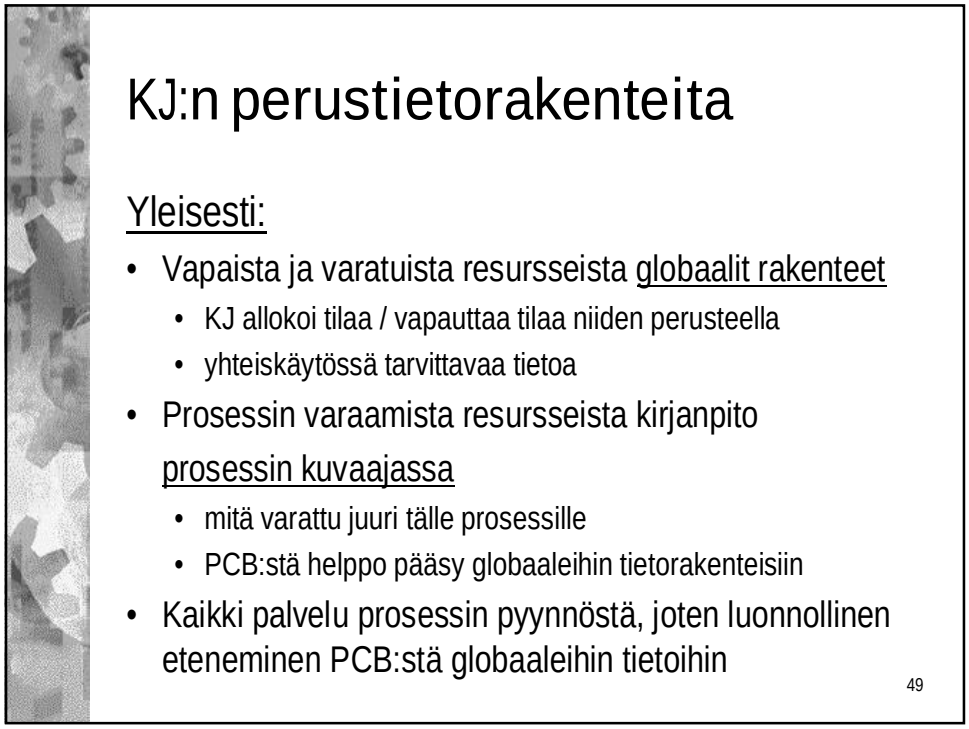
- Tiedostokuvaajat (+levypartitiot)
  - kirjanpito vapaista / varatuista levylohkoista
    - pysyvä kirjanpito levyllä,
      KJ tuo muistiin käsittelyä varten
    - hakemistoalkio per tdsto (myös hsto on tdsto!)
      - mitkä lohkot kuuluvat tdstoon
      - muut tdston attribuutit (mm. omistaja, käyttöoikeudet)
  - kirjanpito avatuista tiedostoista
    - prosessikohtaista PCB:ssä + yhteistä tietoa
    - käyttöoikeuksien tarkistaminen
    - luku / kirjoituspositio
    - yhteiskäyttö: poissulkeminen / synkronointi

47

KJ:n perustietorakenteita

- Laitekuvaajat
  - laitteiden käytössä tarvittavaa tietoa
    - laitteen tunnistus, device id
    - kenelle laite varattu
    - laitteen tila
    - mitä ajuria käyttää
    - mitä ajurin funktiota kutsuttava missäkin tilanteessa
      - open(), read(), write(), close() ...
    - odottavat pyynnöt parametreineen
      - laite palvelee yhtä kerrallaan
- Keskeytys ð mikä ajuri suoritukseen?
- PCB laitteen (ajurin) Blocked-jonossa

48

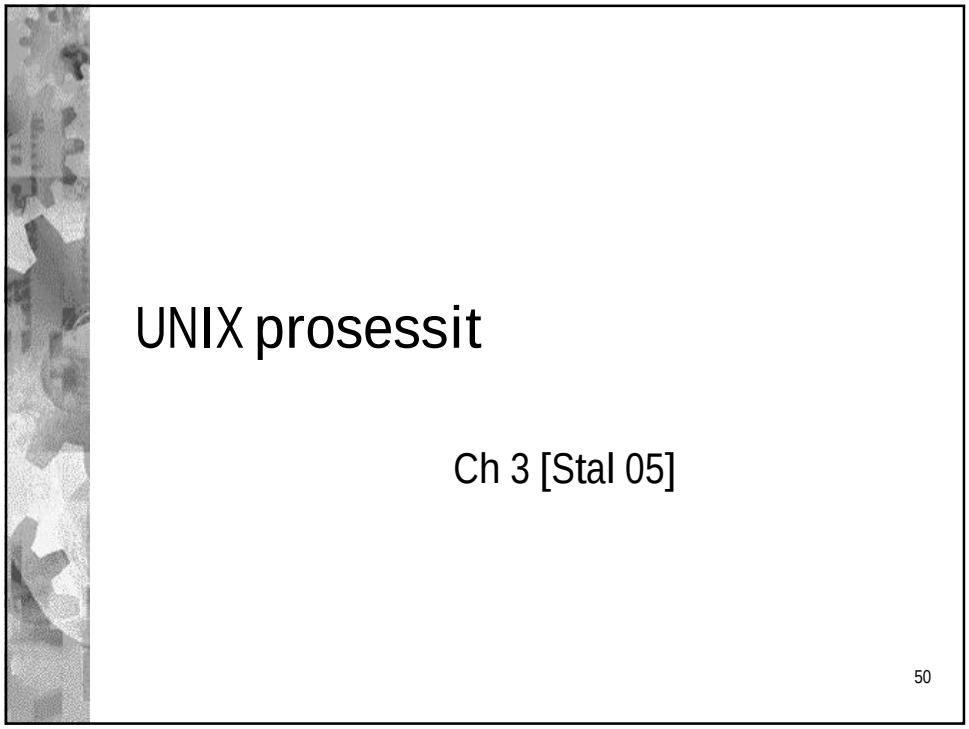


KJ:n perustietorakenteita

Yleisesti:

- Vapaista ja varatuista resursseista globaalit rakenteet
  - KJ allokoi tilaa / vapauttaa tilaa niiden perusteella
  - yhteiskäytössä tarvittavaa tietoa
- Prosessin varaamista resursseista kirjanpito prosessin kuvaajassa
  - mitä varattu juuri tälle prosessille
  - PCB:stä helppo pääsy globaaleihin tietorakenteisiin
- Kaikki palvelu prosessin pyynnöstä, joten luonnollinen eteneminen PCB:stä globaaleihin tietoihin

49



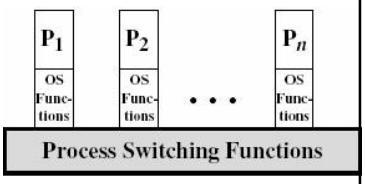
UNIX prosessit

Ch 3 [Stal 05]

50

UNIX SVR4 Prosessit

(Fig. 3.15 (b) [Stal05])



- Pääosa KJ-työstä tapahtuu käyttäjätason prosessien sisällä
  - etuoikeutetussa tilassa (SVC, keskeytys)
- Osa KJ-työstä tapahtuu KJ-prosesseissa
  - etuoikeutetussa tilassa
- Yhdeksän eri prosessin tilaa
- Etuoikeutettu prosessi ei voi menettää suoritinta
  - ellei se itse pyydä jotain resurssia, joka ei ole vapaana
- Prosessi 0 – boot
- Prosessi 1 – init (mother of all)

ks. Fig. 3.17 [Stal05]

51

UNIX: Prosessin tilasiirtymät

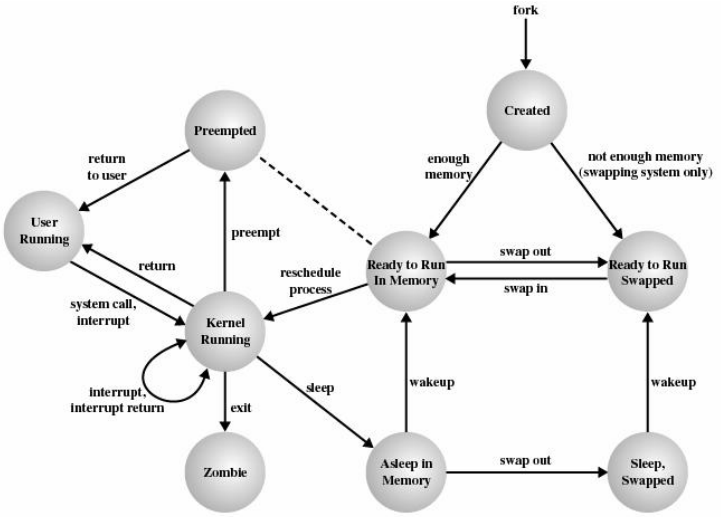


Figure 3.17 UNIX Process State Transition Diagram

52

UNIX prosessin kuvaaja

Tbl 3.10 [Stal05]

- Kääntäjän tiedoista: User Level Context
- Suoritinympäristö: Register Context
- KJ:n tiedot: System Level Context
  - staattinen
    - prosessin tilatiedot (Process Table Entry)

Tbl 3.11 [Stal05]

Tbl 3.12 [Stal05]

    - prosessin hallintotiedot (U Area)
  - dynaaminen
    - muistialueet (Region Table)
    - oma pino etuoik. tilan koodille (Kernel Stack)

53

Table 3.10 UNIX Process Image

User-Level Context	
Process Text	Executable machine instructions of the program
Process Data	Data accessible by the program of this process
User Stack	Contains the arguments, local variables, and pointers for functions executing in user mode
Shared Memory	Memory shared with other processes, used for interprocess communication
Register Context	
Program Counter	Address of next instruction to be executed; may be in kernel or user memory space of this process
Processor Status Register	Contains the hardware status at the time of preemption; contents and format are hardware dependent
Stack Pointer	Points to the top of the kernel or user stack, depending on the mode of operation at the time or preemption
General-Purpose Registers	Hardware dependent
System-Level Context	
Process Table Entry	Defines state of a process; this information is always accessible to the operating system
U (user) Area	Process control information that needs to be accessed only in the context of the process
Per Process Region Table	Defines the mapping from virtual to physical addresses; also contains a permission field that indicates the type of access allowed the process: read-only, read-write, or read-execute
Kernel Stack	Contains the stack frame of kernel procedures as the process executes in kernel mode

54

Table 3.11 UNIX Process Table Entry

Process status	Current state of process.
Pointers	To U area and process memory area (text, data, stack).
Process size	Enables the operating system to know how much space to allocate the process.
User identifiers	The **real user ID** identifies the user who is responsible for the running process. The **effective user ID** may be used by a process to gain temporary privileges associated with a particular program; while that program is being executed as part of the process, the process operates with the effective user ID.
Process identifiers	ID of this process; ID of parent process. These are set up when the process enters the Created state during the fork system call.
Event descriptor	Valid when a process is in a sleeping state; when the event occurs, the process is transferred to a ready-to-run state.
Priority	Used for process scheduling.
Signal	Enumerates signals sent to a process but not yet handled.
Timers	Include process execution time, kernel resource utilization, and user-set timer used to send alarm signal to a process.
P_link	Pointer to the next link in the ready queue (valid if process is ready to execute).
Memory status	Indicates whether process image is in main memory or swapped out. If it is in memory, this field also indicates whether it may be swapped out or is temporarily locked into main memory.

55

Table 3.12 UNIX U Area

Process table pointer	Indicates entry that corresponds to the U area.
User identifiers	Real and effective user IDs. Used to determine user privileges.
Timers	Record time that the process (and its descendants) spent executing in user mode and in kernel mode.
Signal-handler array	For each type of signal defined in the system, indicates how the process will react to receipt of that signal (exit, ignore, execute specified user function).
Control terminal	Indicates login terminal for this process, if one exists.
Error field	Records errors encountered during a system call.
Return value	Contains the result of system calls.
I/O parameters	Describe the amount of data to transfer, the address of the source (or target) data array in user space, and file offsets for I/O.
File parameters	Current directory and current root describe the file system environment of the process.
User file descriptor table	Records the files the process has open.
Limit fields	Restrict the size of the process and the size of a file it can write.
Permission modes fields	Mask mode settings on files the process creates.

56

UNIX prosessin luonti

- Ytimen systeemikutsu *pid=fork()*
  - varaa paikka prosessitaulusta Process Table
  - anna uniikki ID (nelinumeroinen)
  - kopioi vanhemman koko prosessinkuvaaja uudelle prosessille (tai luo ihan uudet rakenteet!)
    - vain linkki yhteiseen muistialueeseen
  - lisää aukiolevien tiedostojen käyttäjien määrää yhdellä (jos tavallinen ”sibling”)
  - siirrä uusi prosessi RR-jonoon
  - palauttaa lapsen pid:n vanhemmalle ja pid=0 lapselle (tämä on ainoa ero koodinäkökulmasta)
    - esimerkki seuraavalla kalvolla

57

UNIX fork-esimerkki

```
while (TRUE) { /* repeat forever /*
        type_prompt( ); /* display prompt on the screen */
        read_command(command, params); /* read input line */
        pid = fork( ); /* fork off a child process */

        if (pid < 0) {
                printf("Unable to fork0); /* error condition */
                continue; /* repeat the loop */
        }

        if (pid != 0) {          /* parent waits for child */
                waitpid (−1, &status, 0);

        } else {                 /* child does the work */
                execve(command, params, 0);
        }
}
```

Fig. 10-7 [Tane01]. A highly simplified shell.

58