

Løsningsforslag INF1060 – H06

Oppgave 1 – operativsystemer – 42 poeng

Løsningsforslagene er mer eller mindre hentet fra foilene. Merk at andre forklaringer kan også være fornuftige.

1a Definisjon (5 poeng)

“An operating system (OS) is a collection of programs that acts as an intermediary between the hardware and its user(s), providing a high-level interface to low level hardware resources, such as the CPU, memory, and I/O devices. The operating system provides various facilities and services that make the use of the hardware convenient, efficient, and safe” – her er det mange forskjellige, men denne inneholder de fleste viktige punktene fra foilene. Pluss hvis de har med extended/virtuel maskin og ressurs manager.

Trenger OS for å håndtere ressurser på en fornuftig måte: rettferdig, effektivt, etc + gjemme/sjule hardware spesifikke detaljer for programmereren (som forenkler en smule!)

1b Systemcall (5 poeng)

Interface mellom user and kernel for å akkessere kjernens tjenester og ressursene den håndterer.

Making a system call is similar to a procedure call, but system calls enter the kernel:

push parameters on stack

call library code

put system call number in register

call kernel (TRAP)

✓kernel examines system call number

✓finds requested system call handler

✓execute requested operation

return to library and clean up

✓increase instruction pointer

✓remove parameters from stack

resume process

1c fork (4 poeng)

*A process can create another process using the pid_t **fork**(void) system call (see man 2 fork) :*

*makes a **duplicate** of the calling process including a copy of virtual address space, open file descriptors, etc... (only PID is different – locks and signals are not inherited)*

returns

- child process' PID when successful, -1 otherwise
- 0 in the child itself when successful

both processes continue in parallel from the same place and state, but can from there go different ways...

1d scheduling (6 poeng)

Threads and processes used to be equal, but Linux uses (in 2.6) thread scheduling

Three classes

✓ SHED_FIFO

- may run forever, no timeslices
- may use it's own scheduling algorithm

✓ SHED_RR

- each priority in RR
- timeslices of 10 ms (quantums)

✓ SHED_OTHER

- ordinary user processes
- uses "nice"-values: $1 \leq \text{priority} \leq 40$
- timeslices of 10 ms (quantums)

✓ Threads with highest goodness are selected first:

- realtime (FIFO and RR): $\text{goodness} = 1000 + \text{priority}$
- timesharing (OTHER): $\text{goodness} = (\text{quantum} > 0 ? \text{quantum} + \text{priority} : 0)$

✓ Quantums are reset when no ready process has quantums left (end of epoch):

$\text{quantum} = (\text{quantum}/2) + \text{priority}$

1e Minneadressering (4 poeng)

✓ Hardware often uses absolute addressing - reading data by referencing the (absolute) byte numbers in memory

✓ Relative addressing

- ✓ independent of process position in memory
- ✓ address is expressed relative to some base location

Relative adressering brukes fordi resultatet av absolutt adressering er avhengig av hvor prosessen plasseres i minnet og dette er forskjellig hver gang...

1f I/O i Windows (6 poeng)

I/O manager perform caching centralized facility to all components (not only file data)

I/O requests processing:

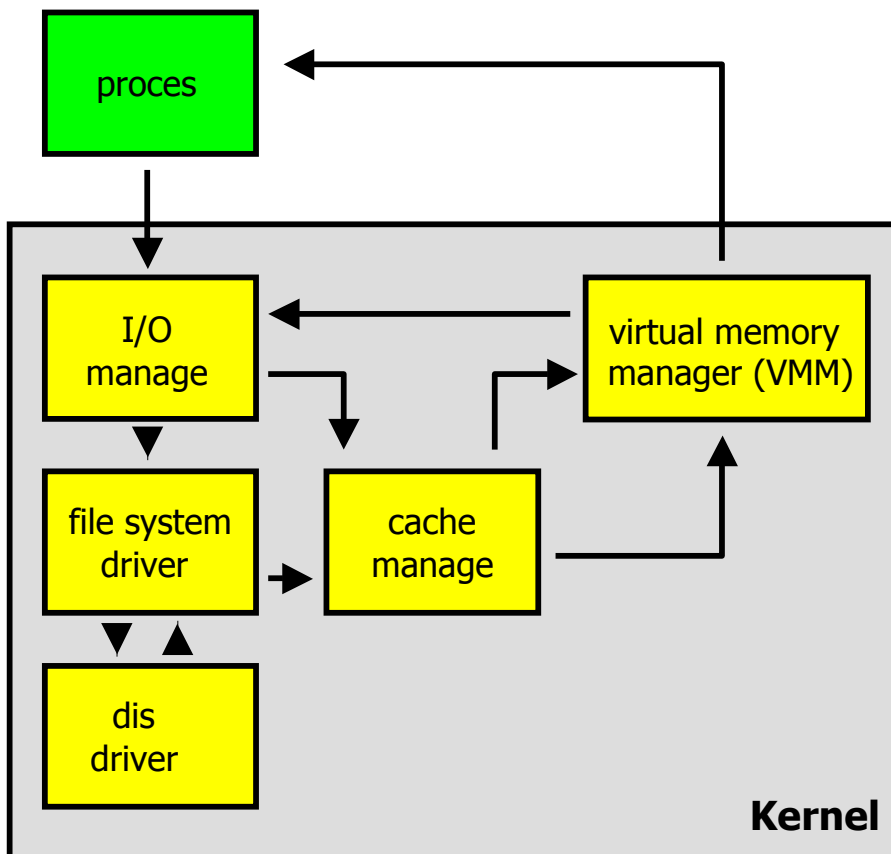
- 1. I/O request from process*
- 2. I/O manager forwards to cache manager*

in cache:

- 3. cache manager locates and copies data to process buffer via VMM*
- 4. VMM notifies process*

on disk:

- 3. cache manager generates a page fault*
- 4. VMM makes a non-cached service request*
- 5. I/O manager makes request to file system*
- 6. file system forwards to disk*
- 7. disk finds data*
- 8. reads into cache*
- 9. cache manager copies data to process buffer via VMM*
- 10. virtual memory manager notifies process*



1g sidefeil (6 poeng)

En sidefeil oppstår når man prøver å aksessere en side/page som ikke er i minnet. Under er håndteringen vi snakket om i forelesning. De bør ha fått med seg de viktigste punktene:

- 1. Hardware traps to the kernel saving program counter and process state information*
- 2. Save general registers and other volatile information*
- 3. OS discover the page fault and tries to determine which virtual page is requested*
- 4. OS checks if the virtual page is valid and if protection is consistent with access*
- 5. Select a page to be replaced*
- 6. Check if selected page frame is "dirty", i.e., updated*
- 7. When selected page frame is ready, the OS finds the disk address where the needed data is located and schedules a disk operation to bring in into memory*
- 8. A disk interrupt is executed indicating that the disk I/O operation is finished, the page tables are updated, and the page frame is marked "normal state"*
- 9. Faulting instruction is backed up and the program counter is reset*
- 10. Faulting process is scheduled, and OS returns to routine that made the trap to the kernel*
- 11. The registers and other volatile information is restored and control is returned to user space to continue execution as no page fault had occurred*

Pluss hvis de har noe om hvordan det for eksempel skjer på Pentium: adresse i CR2, lese page directory, så page table og så page. Regner ikke med at de husker de forskjellige bittene i de andre CR.

1h lagring (6 poeng)

✓ *Bruker noe lignende extents i linux hvor en peker brukes for å adressere flere blokker (= runs)*

- ✓ *Each partition contains a master file table (MFT)*
- a linear sequence of 1 KB records*
 - each record describes a directory or a file (attributes and disk addresses)*
 - first 16 reserved for NTFS itself*

A file can be ...

- stored within the record (immediate file, < few 100 B)*
- represented by disk block addresses (which hold data) in the record runs of consecutive blocks (<addr, no>, like extents)*
- use several records if more runs are needed and the element in the record is then the index in the MFT to a new record*

Oppgave 2 - kommunikasjon— max 40 poeng

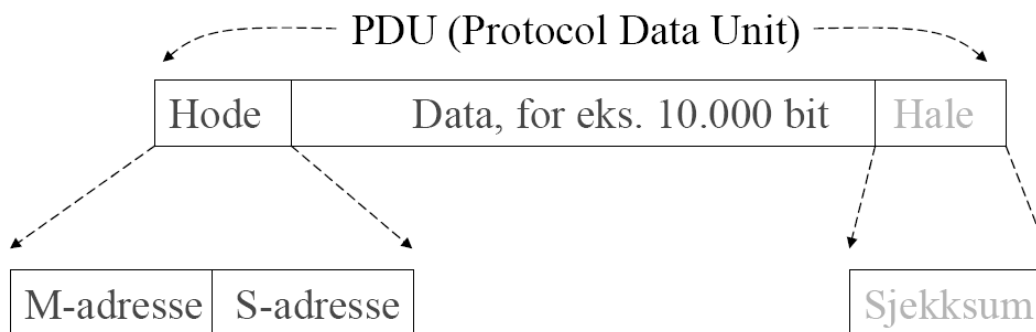
Løsningsforslagene er mer eller mindre hentet fra foilene. Merk at andre forklaringer kan også være fornuftige.

2a Hva er Internet? (3 poeng) – *Internet er et nettverk av nettverk, dvs. mange proprietære, regionale og offentlige nett knyttes sammen via rutere i “kjernenettverket”. Internet er delvis hierarkisk.*

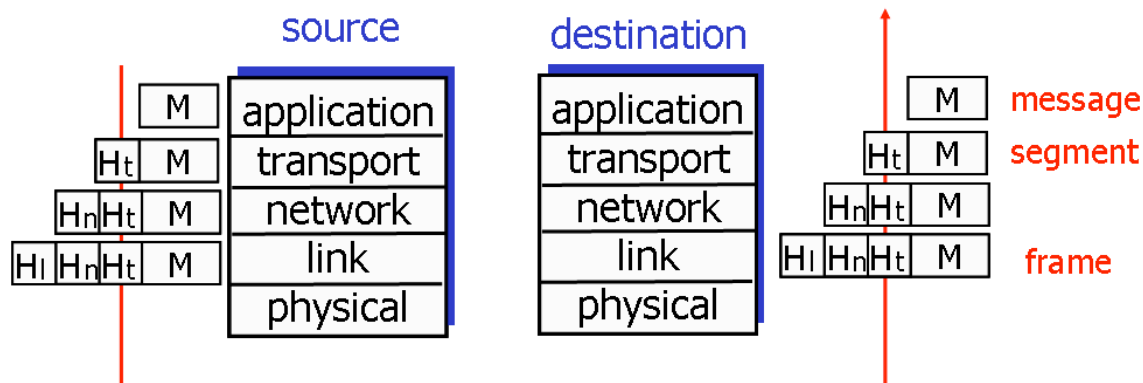
2b Referansemodeller. (5 poeng) - *De omtalte referansemodellene er lagdelte (hierarkiske). Hovedgrunnen til at de har blitt spesifisert er at kommunikasjonssystemer er meget komplekse, og det viste seg å være nødvendig å modularisere. Dette forenkler bl.a. design, vedlikehold og oppdatering av et system. Lagdelingen gjør det mulig å samle og identifisere ulike funksjonelle egenskaper i ulike lag og å spesifisere relasjoner mellom dem. Intensjonen er også at ulike implementasjoner av et lag skal kunne fungere sammen med alle implementasjoner av andre lag (dvs. det er funksjonelle modeller). > En god besvarelse bør inneholde en beskrivelse av hovedfunksjonene til hvert lag. Hovedforskjellen mellom TCP/IP og OSI modellen er at førstnevnte mangler Sesjons- og Presentasjonslag. Dette innebærer at Applikasjonene i Internet selv må ivareta de funksjonene som disse lagene utfører (må være innebakt i applikasjonsprotokollene).*

2c Kontrollinformasjon. (5 poeng)

Generelt pakkeformat (rammer, pakker, meldinger):



Som vist på figuren nedenfor, får hvert lag på sendersiden data fra det laget over, adderer header informasjon for å generere en ny protokoll data enhet, PDU, (melding, segment, ramme, pakke...) som sendes til laget under. På mottagersiden stripes header-informasjonen bort etter hvert som PDU-ene flytter oppover. Det vil ofte også være kontroll-informasjon i en "hale" bakerst i PDUene som behandles tilsvarende.



2d Det fysiske laget (4 poeng). – Grunnen til at man tvinner trådene i trådpar er for å minske virkningen av både ekstern elektromagnetisk påvirkning og elektromagnetisk påvirkning mellom ledningene selv. I det første tilfellet sørger man for at forskjellen i spenning er den samme alltid slik at man tolker signalene riktig hvis man har en positiv og negativ spenning på hver tråd. I tillegg vil feltene som genereres for hver tråd oppheve hverandre.

Andre medier: Her bør koaksialkabel, fiberoptisk kabel og radiolinker tas med. Følgende kan sies: - Koaks: Bra: God skjerming mot el.magnetisk støy; kan handtere høye hastigheter; lett tilkopling av nye stasjoner (on the fly – uten driftsavbrudd). Ulempe: stiv - kan være vanskelig å legge. - Fiber: Meget høye hastigheter; lang rekkevidde; ikke påvirket av el.magnetisk interferens. Ulempe: koplisert å “skjøte”; dyrt tilkoplingsutstyr til maskiner; brukes primært mellom rutere og servere. - Radio: (Mange ulike typer) Bra: fleksibelt - kan f.eks. brukes i ulendt terreng; satelitt-kommunikasjon åpner for langdistanse overføring; forholdvis høye hastigheter; Ulempe: “Åpent” nettverk - alle kan lytte på linkene. Stor feilsannsynlighet pga. refleksjoner, blokkering (bygninger) og vær.

2e Fysiske egenskaper ved overføring (5 poeng) – Tiden det tar er summen av tiden det tar å flytte signalet over kanalen (propageringstiden) og tiden det tar å putte bitene ut på kanalen. Her er.

$$\text{Propageringstiden} = 1000 \cdot 1000 \text{m} / 300.000.000 \text{m/s} = 3.3333 \cdot 10^{-3} = 3.333 \text{ms}$$

$$\text{Tid for å putte bit på } 10 \text{Mbit/s kanalen} = 4000 \text{bit} / 10 \cdot 10^6 \text{bit/s} = 4 \cdot 10^{-4} \text{s} = 0.4 \text{ms}$$

$$\rightarrow \text{total overføringstid} = 3.333 \text{ms} + 0.4 \text{ms} = 3.733 \text{ms}$$

$$\text{Tid for å putte bit på } 10 \text{Gbit/s kanalen} = 4000 \text{bit} / 10 \cdot 10^9 \text{bit/s} = 4 \cdot 10^{-7} \text{s} = 0.0004 \text{ms}$$

$$\rightarrow \text{total overføringstid} = 3.333 \text{ms} + 0.0004 \text{ms} = 3.3337 \text{ms}$$

2f Linklaget (3 poeng) – Kandidatene bør kjenne til og beskrive minst en av følgende innrammingsmetoder:

- Tegn-telling.
- Flag bytes med bytestøffing.
- Start og slutt flag bytes med bitstøffing.

2g Svitsjetechnikker (4 poeng) – Følgende egenskaper gjelder for bruk av henholdsvis linjesvitsjing og pakkesvitsjing:

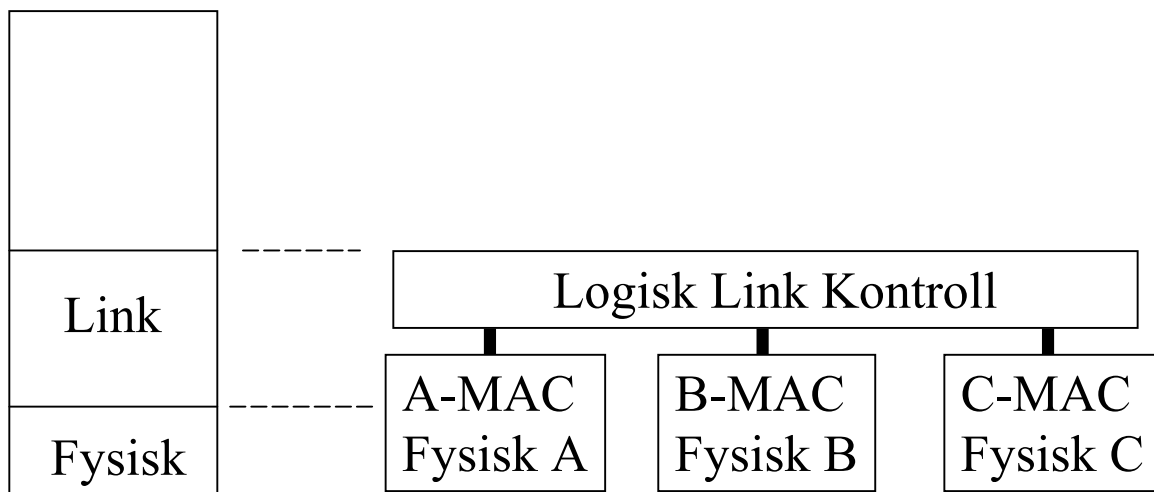
- ❑ *Linjesvitsjing*
 - maks 10 brukere
 - Sannsynlighet for tap: 0%
 - Ubrukt kapasitet: ~90%
- ❑ *Pakkesvitsjing*
 - >10 kan være active samtidig!
 - Sannsynlighet for tap >0%
 - Ubrukt kapasitet: < 90%

2h Broer vs. Ruter (4 poeng) –

Broer; knytter sammen lokalt nett på link-nivå. Framsendingen er basert på MAC-adresser, hvilket utgjør et effektivt sammenkoplingsalternativ. Broer kan benyttes for isolering av trafikk (til segmenter), og konsumerer ikke IP-adresser.

Rutere; foretar derimot framsending basert på IP-adresser, og sørger for å rute pakker (på Nettlaget) gjennom nettet fra en kildemaskin til målmaskin. Hovedkomponentene i en ruter er inn og utkøer, rutetabell samt preprosseserings-, ruting- og framsendingsprosesser. Preprosseserings-prosessen plukker pakker fra inn-køen og leverer adresse-info (samt linjenummer) til ruting-prosessen. Denne oppdaterer rutetabellen vha. den mottatte informasjonen, og overlater pakka til framsendings-prosessen som legger pakka i riktig utkø.

2i Lokalt nett (4 poeng) – Topologier som minimum bør nevnes er bus, fysisk ring, logisk ring og radio-LAN (uten struktur). Videre bør man forklare essensen ved følgende figur:



2j Nettlaget (3 poeng) – *Definisjonen av multicast sier at det er snakk om 1:n type kommunikasjon, dvs. fra en avsender sendes det til n mottakere. Man vil med denne teknikken sende til en gruppe av mottakere vha. en enkelt sending i stedet for en sending pr. mottaker. Dette vil minske loaden både på nettverket og senderen. Listen av mottakere kan endres over tid; dette kan for eksempel administreres av avsenderen.*

Oppgave 3 – flervalg 18 poeng

Riktig = 2 poeng

Galt = -1 poeng

Oppgave 3a: *Riktige svar: 3, 4, 7, 8, 11*

Merk:

4 er ment å være riktig, men svaret kan leses slik at kjøringen av OSet også er med. Slik var det ikke opprinnelig tenkt, og noen definerer ikke dette en del av boot. De som stilte spørsmål om dette ved eksamen fikk beskjed om å redegjøre hva de leste med "laste". Så legg dette til grunn når dere retter.

10 kan også missforsås, dvs. At man kan lese det slik at den scheduleres ut og inn, men kan kjøre "evig". Det er tenkt slik at vi ikke har preemption, noe som vi har i Windows → feil

Oppgave 3b: *Riktige svar: 1, 3, 6, 7*