

INF1060: avsluttende eksamen

Kandidatnummer: _____

1 Oppgave 1 - operativsystemer — *max 40 poeng*

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

❑ 1a (16 poeng) –

❑ 1a - 1 (2 poeng) –

Et program er bare settet av instruksjoner.

En prosess er da utførelsen/eksekveringen av et program med tilstand, variable, etc. som må til for å utføre alle instruksjonene.

❑ 1a - 2 (4 poeng) –

The `int execve(const char *filename, char *const argv[], char *const envp[])` executes the program pointed to by filename in the environment (environment variables) specified by envp using the arguments passed by argv.

In line 18, `/bin/ls` is executed using the parameters `-l` and `/bin` (i.e., long/detailed list of directory contents in the `/bin` directory) in the environment specified by the users environment variables pointed to by `environ`.

❑ 1a - 3 (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: 1.) child process' PID when successful, -1 otherwise ; 2.) 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.

❑ 1a - 4 (6 poeng) –

In short, the program spawns two processes using `fork`. Assuming everything goes OK:

The parent process 1) writes the PID of itself and the child, respectively, in line 11 ; 2) waits for a (any) child to finish and puts the returned status in `n` ; 3) prints the returned childs PID and status in line 13.

The child 1) fills the `argv` array ; 2) prints the PID of itself and "0" in line 17 ; 3) executes `ls` as described above ; 4) and will NEVER return 999 (unless `execve` fails).

❑ 1b (13 poeng) –

❑ 1b - 1 (3 poeng) –

Relative memory addressing is used because processes move in memory between each execution of the program - thus, accessing absolute address `0xfff...111` will give different results each time.

Relative addressing is independent of process position in memory. The address is expressed relative to some base location, and dynamic address translation - find absolute address during run-time adding relative (set to position within program) and base (set to beginning of a programs memory when started) addresses.

❑ 1b - 2 (5 poeng) –

Se fig. 1. Boka nevner text, data og stack segment som de bør kunne faa med seg: 1) Text (code) segment holder program instruksjonene og er statisk i storrelse, 2) data segmentet inneholder globale variable og dynamisk minne

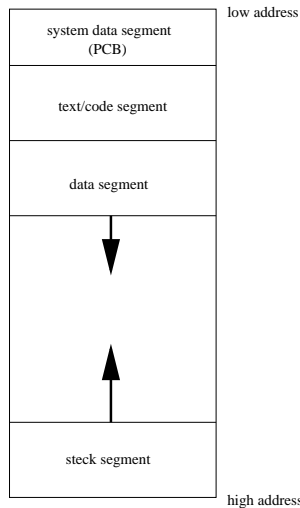


Figure 1: Process memory layout

(f.eks. allokert med malloc) og er dynamisk (vokser mot hoyere minne addresser), og 3) stack segmentet inneholder ofte funksjonsvariable, lagrede register tilstander, og er dynamisk (vokser mot lavere minne addresser). I tillegg har vi snakket litt om PCB som kan legges i starten, men dette er bare paa foiler.

❑ 1b - 3 (5 poeng) –

Virtuelt minne er en mekanisme for å gi inntrykk av at man har mer minne tilgjengelig en det som er fysisk satt inn i maskinen. Skulle hjelpe på problemene med å partisjonere programmer, etc. fordi man tidligere hadde lite minne. Observasjonen var da at man ikke trengte alle data hele tiden.

Programmene blir her delt i mindre deler (like store rammer kaldt sider/pages) hvor da noen er fysisk i minnet mens andre er lagret på disk og hentes ved behov.

En virtuell adresse oversettes til en fysisk adresse ved hjelp av en side/pagetabel i MMUen: Overste bittene i virtuelle adressen brukes som offsett/indeks til pagetabellen. I denne raden i tabellen finnes informasjon om den fysiske siden. Man har minst en adresse og et "present bit": hvis bittet er 0 er siden på disk og adressen sier ofte noe om disk blok adressen, men er bittet 1 er siden minnet og adressen gir de adressen til siden i minnet. For så å finne byten vi leter etter brukes de nederste bittene i den virtuelle adressen (typisk de 12 nederste for 4KB sider) som offsett i siden.

❑ 1c (6 poeng) –

❑ 1c - 1 (3 poeng) –

Access time = seek + rotational delay + transfer time (+ "other")

However, "other" is usuallt negligible. Thus, I expect the three first to be mentioned.

❑ 1c - 2 (3 poeng) –

SCAN sweeps across the disk and serves request according to current position over the platter and the direction it moves, i.e., moves head from edge to edge and serves requests on the way. It is bi-directional and is a compromise between response time and seek time optimizations.

❑ 1d (5 poeng) –

Pipes is a classic IPC method used for processes to communicate and send data. `int pipe(int filedes[2])` creates a pair of file descriptors, pointing to a pipe inode, and places them in the array pointed to by filedes. filedes[0] is for reading, filedes[1] is for writing.

```
> ls -l | more:
```

- shell runs two processes ls and more which are linked via a pipe
- the first process (ls) writes data (e.g., using write) to the pipe
- the second (more) reads data (e.g., using read) from the pipe

2 Oppgave 2 - Datakom — max 39 poeng

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

❑ 2a (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 (3 poeng) –

Forbindelsesløs overføring: Kalles også datagramtjeneste. Hver pakke bærer nødvendig adresseinfo til å finne fram til mottaker samt sin egen adresse. Rutes individuelt gjennom nettet og ruterne forsøker å finne en mest mulig optimal vei. Dette er hovedfordelen. Ulempen er at det ikke gis noen garantier ("best effort") for at pakker kommer fram, forsinkelser og tap av pakker kan oppstå. Forbindelsesorientert overføring: Følgende skjer: Oppkopling, dataoverføring, nedkopling. Opp- og nedkoplingen skal sikre at mottatte data ikke misstolkes (f.eks. at stasjonen mener at de hører til en annen forbindelse). I datafasen vil dataene komme i samme sekvens som de ble sendt og feilfritt. Mottaker kan signalisere at bufferne er fulle (flytkontroll).

❑ 2c (5 poeng) –

De omtalte referansem modellene 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).

❑ 2d (5 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: komplisert å "skjote"; 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 (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:

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

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

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

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

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

❑ 2f (5 poeng) –

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

❑ 2g (4 poeng) –

(Broer:) Knytter sammen lokalnett 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

CSMA/CD (IEEE 802.3)

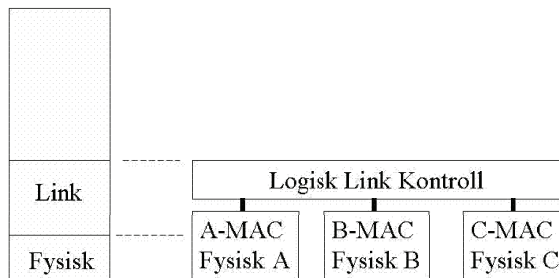


Figure 2: MAC i OSI-stacken

fra en kildemaskin til målmaskin. Hovedkomponentene i en ruter er inn og utkøer, rutetabell samt preprosesserings-, rutings- og framsendings-prosesser. Preprosesserings-prosessen plukker pakker fra inn-køen og leverer adresse-info (samt linjenummer) til rutings-prosessen. Denne oppdaterer rutetabellen vha. den mottatte informasjonen, og overlater pakka til framsendings-prosessen som legger pakka i riktig utkø.

2h (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.

2i (3 poeng) –

UDP: Ingen forbindelser, upålitelig, uordnet, kan miste pakker, pakke-orientert (data en bruker sender sendes som en pakke), ingen 'flyt-kontroll' (sender så fort som mulig), ingen 'congestion control' som vil si at man ignorerer hva som skjer i nettverket.

TCP: Forbindelse opprettes ved '3-veis handshake'/nedkobles ved 'teardown', tilstandsinformasjon, pålitelig (mister ikke pakker vha. retrans.), leveres i samme rekkefølge, strømorientert (bufrer opp data og sender en pakke når det ser 'fornuftig' ut, dvs en sendt mengde data fra applikasjonen sendes ikke nødvendigvis ut som en enkelt pakke), har flyt kontroll ved at man ikke sender fortere enn mottaker kan motta, 'congestion control' ved at vi adapterer til problemer i nettverket.

2j (3 poeng) –

Kommunikasjonen over nettet mellom distribuerte applikasjons-prosesser må foregå i en (standardisert) syntaks som begge sider oppfatter på samme måte. En slik syntaks kaller vi 'overføringssyntaks'. Denne er nødvendig fordi kommunikasjonen foregår mellom inhomogene ende-systemer (eks. ulik maskinvare, ulike OS, ulike programmeringsspråk).

3 Oppgave 3 - flervalgsoppgaver — max 21 poeng

Hvert riktige svar gir 1 poeng.

3a – OS (10 poeng): 1, 3, 7 og 9 er riktige

3b – Datacom (11 poeng): 4, 6, 8, 10 og 11 er riktige

► Poeng totalt: _____ (max 100 poeng)