

INF1060: avsluttende eksamen

Kandidatnummer: _____

□ Oppgave 1 - operativsystemer — *max 40 poeng*

- 1a (**5 poeng**) – BOKA: sett med system SW rutiner som sitter mellom applikasjone og HW, som definerer et set av standard interfaces, tilbyr tjenester og virker som en platform mellom for å kjøre og utvikle programmer. I TILLEGG: bør de få med seg nøkkelor-dene (eller tilsvarende begreper) “virtuel maskin” (som skal skjule komplekse detaljer om underliggende HW, enkel å bruke, “inntrykk av en maskin for hver prosess”, ...) og ressurs-manager” (som ska tilby forskjellige muligheter for å bruke HW på en fornuftig, effektiv, rettferdig og sikker måte)
- 1b (**5 poeng**) – BOKA: 1) process: a) windows - objekt som består av et eksekverbart program, b) unix - eksekvering av et image. 2) program: set med instruksjoner som forteller maskinen hva slags operasjon dne skal utføre. PÅ FORELESNING: analogien mellom kokebok (program) og kokk (prosess) som lager mat fra en oppskrift hvor man har tilstand etter hvor langt man har kommet.
- 1c (**10 poeng**) – Windows XP/2000 har en preemtiv scheduler hvor hver tråd får tildelt processor tid gitt i quantum. Hvis en tråd går tom for quantum, avbrytes den. Prioritet-basert, dvs. en hvis en prosess med høyere prioritet kommer får den kjøre (hvis qunatums igjen). Mange prioritetsnivåer (32, 31 - 16 system / 15 - 1 user prosesser + 0 idle thread - boka sier 15 - 0 er user prosesser). Hver bruker prosess har en base prioritet, men denne kan justeres ut i fra bruk: bruker mye → prioriteten blir lavere / mye I/O eller interaktivitet → høyere prioritet
- 1d (**5 poeng**) – context switches er en multitasking mekanisme for å bytte kjørende prosesser, dvs. operasjonen med å bytte fra en prosess’ context (lagre tilstandsverdier som register verdier som f.eks. legges på stacken) til en annen prosess’ context (hvor dens register verdier (og annen tilsdandsinformasjon) leses fra stacken).
- 1e (**9 poeng**) – se fig. 1, merk at de to forespørselene på 15 ikke medfører flytting i SCAN.
- 2f (**6 poeng**) – se fig. 2. Boka nevner text, data og stack segment som de bør kunne få med seg: 1) Text (code) segment holder program instuksjonene og er statisk i størrelse, 2) data segmentet inneholder globale variable og dynamisk minne (f.eks allokert med malloc) og er dynamisk (vokser mot høyere minne addresser), og 3) stack segmentet inneholder ofte

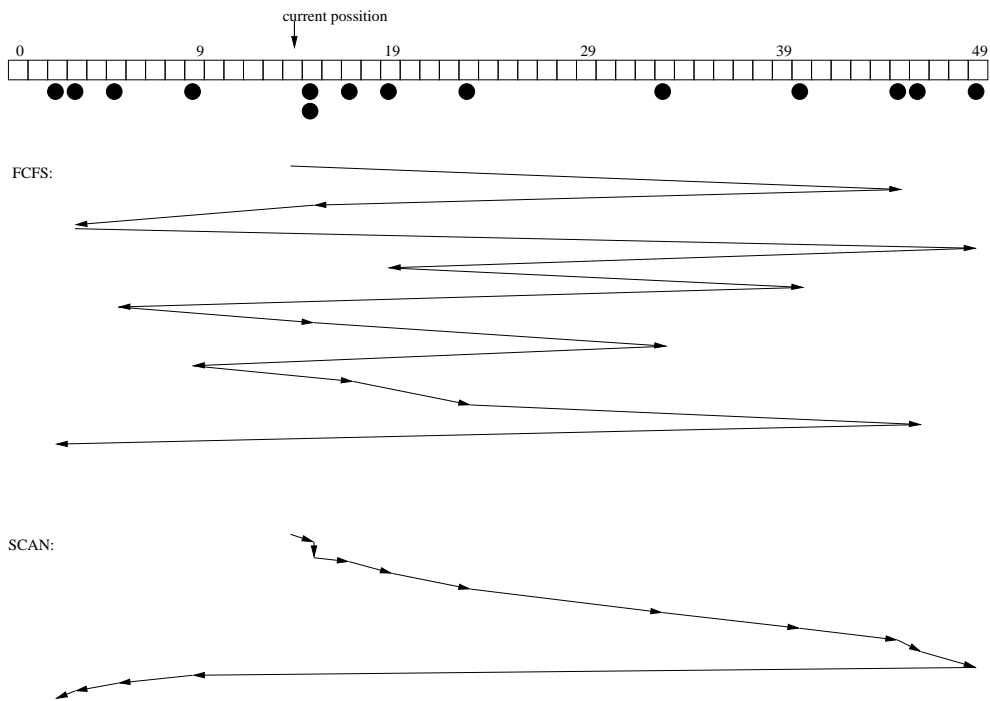


Figure 1: Disk arm movement for FCFS / SCAN

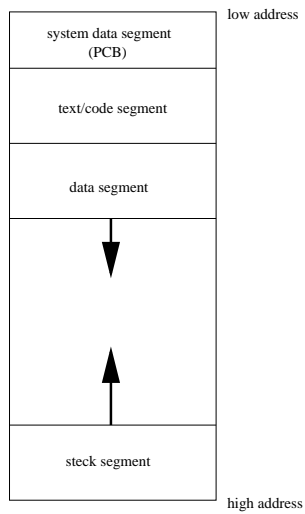


Figure 2: Process memory layout

funksjonsvariable, lagrede register tilstander, og er dynamisk (vokser mot lavere minne adresser). I tillegg har vi snakket litt om PCB som kan legges i starten, men dette er bare på foiler.

□ **Oppgave 2 - kommunikasjon** — *max 30 poeng*

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

□ **2a (14 poeng)** fordelt under

1. **(4 poeng)** Protocols define formats, order of sending and receiving of messages, and the actions that the reception initiates.

Protokollstakk: nettverket er komplekst, dele opp funksjonalitet for å få struktur. Modularisering forenkler.

2. **(4 poeng)** Poenget her er at de viser hvordan innkapsulering virker. Dvs, hvis vi har TCP, IP, Ethernet vil det se noe slikt ut:

| Ethernet | IP | TCP | applikasjonsdata | (og evt. ekstra ethernet framing preamble/tail)

3. **(6 poeng)** UDP: ingne forbindelser, upålitelig, ouordnet, kan miste pakker, pakkeorientert (data en bruker sender sendes som en pakke), ingen “flow control” (sender så fort som mulig), “ingen congestion control” som vil si at man ignorerer hva som skjer i nettverket.

TCP: forbindelse opprettes ved “handshake”/nedkobles ved “teardown”, tilstandsinformasjon, pålitelig (mister ikke pakker vha retrans.), leveres i samme rekkefølge, strømorientert (bufferer 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 en mottaker kan motta, “congestion control” ved at vi adapterer til problemer i nettverket.

- **2b (10 poeng)** – forbindelsesløs: ikke forbindelses oppsett, ingen forberedelse til sending, mottaker kan motta når som helst, ingen tilstand i maskinene, sender vet ikke om noen venter på meldinger, må identifisere meldingene selv, ingen nedkobling

forbindelsesorientert: oppsett av forbindelse nødvendig (handshake), forberedelser av sending, tilstander i maskinene, endepunktene kjenner hverandre, vet hvilken forbindelse en pakke/melding kommer fra, må ha en “teardown”

- **2c (6 poeng)**, fordelt under – antar megabit = $(10^3)^2$ bit = 1.000.000 bit, men også $2^{32} = 1.048.576$ er riktig (hvis noen har tenkt riktig, men feil svar, bør de få noe uttelling)

1. **(2 poeng)** Tid for å putte data på kanalen = $8000 \text{ bit} / 2.000.000 \text{ bit/s} = 4 \text{ ms}$
tid får overføring = $6000 \text{ km} / 300.000 \text{ km/s} = 20 \text{ ms}$
totalt = $4 + 20 = 24 \text{ ms}$
2. **(2 poeng)** totalt = $(8000 / 155.000.000) + (6000 / 300.000) = 20,05 \text{ ms}$
3. **(2 poeng)** totalt = $(8000 / 155.000.000) + (10 / 300.000) = 0,085 \text{ ms}$ (85 microsekunder)

□ **Oppgave 3 - C programmering** — *max 10 poeng*

Her må vi være ops på noe som har falt ut av teksten (programmet). Linjen med deklarasjon av f d og x mangler, slik at de både sier noe om dette og gir fornuftig forklaringer **og** de som løser det som tenkt under, må alle få poeng!!!.

- 1 (**2 poeng**) – **rett**: Variablen `y` er initialisert til 0 og økes hele tiden med antall bytes lest (`n` som får sin verdi fra `read`).
 - 2 (**3 poeng**) – **galt**: Vi har en `malloc` i `while`-løkken uten at vi frigjør noe. Minnelekasjen vil tilslutt krasje programmet.
 - 3 (**2 poeng**) – **rett**: ... (men `malloc` kan også gå galt)
 - 4 (**3 poeng**) – **galt**: Skrivningen til `fil` i linje 17 vil alltid være ulovlig fordi `filen` er åpnet bare for lesing - vil alltid utføre `exit(2)`.
- Oppgave 4 - flervalgsoppgaver** — *max 20 poeng*
hvert riktige svar gir 2 poeng, hvert gale -1 poeng, dvs. gi et poeng for hver riktige påstand de har trukket ut, og trekk et poeng hvis de har trukket ut et galt (hadde ikke tenkt at det er minus hvis de ikke trekker ut/glemmer et riktig svar).
- 4a – (**10 poeng**) minne: riktige alternativer: 1, 3, 4, 5 og 7
 - 4b – (**6 poeng**) nettverkslaget: riktige alternativer: 2, 3 og 5
 - 4c – (**4 poeng**) C-programmering: riktige alternativer: 2, 3 og 5 (5 er egentlig litt missvisende, dvs. hvordan det er spurt. I utgangspunktet tenkte jeg at dette skulle være feil, men denne tekststrengen blir jo skrevet ut - det er bare det at det er mer som skrives ut!!! Jeg vil derfor anbefale å la 5 være riktig, men ikke trekke hvis noen svare FALSE).

► **Poeng totalt:** _____ (*max 100 poeng*)