

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF1060 — Introduksjon til operativsystemer og datakommunikasjon
Eksamensdag:	8. desember 2004
Tid for eksamen:	14.30 – 17.30
Oppgavesettet er på	5 sider.
Vedlegg:	Ingen
Tillatte hjelpemidler:	Kalkulator

Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene.

### Oppgave 1 Operativsystemer (ca. 40 %)

Svar *kort og presist* på følgende oppgaver om operativsystemer.

#### 1a Definisjon og oppgaver

Hva er et operativsystem, og hva er dets oppgaver?

#### 1b Prosesser

Hva er forskjellen på et program og en prosess?

#### 1c CPU scheduling

Forklar hvordan scheduleren på Windows XP/2000 virker!

#### 1d Context switch

Hva er en “context switch”?

#### 1e Disk Scheduling

Anta at vi har en (veldig liten) disk som har 50 spor (“tracks”), nummerert fra 0 til 49, som disk-hodet beveger seg over og leser data fra. Anta at på et bestemt tidspunkt leser disken en sektor fra spor 14. Etter denne forespørselen ligger det følgende forespørsler i køen (hvert tall henviser til hvilket spor den forespurte blokken ligger på, og ordningen viser i hvilken rekkefølge forespørselene ankom systemet med forespørselen lengst til høyre, dvs. 2, som første):

45 15 3 49 19 40 5 15 33 9 17 23 46 2

Vis i en grafisk figur hvordan diskhodet beveger seg over de forskjellige sporene (i hvilken rekkefølge) hvis vi bruker henholdsvis “First-Come-First-Serve” (FCFS) og SCAN (diskhodet er på vei fra lavt-nummererte spor til

(Fortsettes på side 2.)

høyt-nummererte spor) og alle forespørslene i køen kan betjenes i samme runde.

### 1f Minne

En prosess deler opp sitt eget minne hvor hver del (hvert segment) brukes på forskjellige måter/til forskjellige ting. Hvordan gjøres dette, og hva brukes hver del/segment til?

## Oppgave 2 Datakommunikasjon (ca. 30 %)

Svar *kort og presist* på følgende oppgaver om datakommunikasjon.

### 2a Protokoller

1. Hva er en protokoll, og hva er poenget med en protokollstakk?
2. Protokollene i TCP/IP-stakken genererer protokoll-hoder ("protocol headers") som de legger til de dataene de skal sende. Vis hvordan et dataelement som kommer gjennom socket-laget pakkes inn og hvordan det legges til protokoll-hoder.
3. Hvilke transportlagsprotokoller er definert for TCP/IP-stakken, og hva er de viktigste egenskapene med dem?

### 2b Forbindelsesløse- og forbindelsesorienterte protokoller

Hva er forskjellen på forbindelsesløse- og forbindelsesorienterte protokoller?

### 2c Transmisjonsforsinkelse

Anta at du er aksjemegler og får oppdateringer fra børsen i New York cirka 6000 km unna en gang i sekundet i en pakke på 1000 bytes. Anta videre at signalene går med lysets hastighet (rundet av til 300.000 kilometer i sekundet) og at vi kan se bort i fra forsinkelsen i mellomnodene.

1. Hvor lang tid tar det fra pakken sendes fra New York til du har fått oppdateringene her i Oslo (hva er forsinkelsen) hvis båndbredden på datakanalen er begrenset av din kanal på 2 megabit per sekund (Mbps) som for eksempel en ADSL-linje?
2. Du finner fort ut at det å kjøpe aksjer til en god pris ofte handler om å bestemme seg raskt og syntes derfor at forsinkelsen er for stor (!). Du bestemmer deg derfor for å skifte ut ADSL linjen med en fastlinje som har en 155 Mbps datakanal. Hva er den nye forsinkelsen nå?
3. Hva er forsinkelsen hvis du begynner å handle på samme måte over nettet (med den nye fastlinjen) på Oslo børs (10 km unna) i steden for børsen i New York?

(Fortsettes på side 3.)

---

```
1: int A(void)
2: {
3:     int size = 1000000, n, y;
4:     void *p;
5:
6:     y = 0;
7:     if ((fd = open("my.file", O_RDONLY , 0)) == -1) exit(1);
8:
9:     while (TRUE) {
10:         p = malloc(size);
11:         if ((n = read(fd, p, size)) > 0) {
12:             y += n;
13:         }
14:         else {break;} /* LEAVE WHILE LOOP */
15:     }
16:
17:     if ((x = write(fd, p, size)) == -1) exit(2);
18:     close(fd);
19:     return 100;
20: }
```

---

Figur 1: Funksjon A

### Oppgave 3 C programmering (ca. 10 %)

Anta at du har den koden beskrevet i figur 1 fra en funksjon i et C program. Bedøm om de følgende utsagn er riktige eller gale, og begrunn med *en setning eller to* hvorfor/hvorfor ikke.

1. Etter tilordningen `y += n` (linje 12) vil variabelen `y` inneholde verdien på hvor mange bytes som totalt har blitt lest fra filen *my.file* hittil.
2. Koden kan kjøre evig hvis vi antar at filen *my.file* som leses er lang nok (uendelig stor).
3. Etter `malloc` peker pekeren `p` på et allokert minneområde som er 1000000 bytes stort.
4. Hvis vi leser slutten av filen (har lest hele *my.file*) vil koden etterhvert avslutte og funksjon returnere med statusverdi 100 (linje 19).

### Oppgave 4 Diverse flervalgsoppgaver (ca. 20 %)

Denne oppgaven er en oppgave hvor du skal velge ett eller flere svar alternativer (flere kan være riktig) for hver oppgave (flervalgsoppgaver) uten å gi begrunnelse. Hvert riktig svar gir poeng, og hvert gale svar gir minuspoeng.

(Fortsettes på side 4.)

#### 4a Minne

Hvilke utsagn er riktige:

1. En prosess har ingen garantier for at den plasseres på samme sted i minnet hvis vi kjører det flere ganger
2. En relativ minneadresse gir fysisk posisjon i minnet
3. For å finne et data element på en bestemt fysisk adresse under kjøring av en prosess må vi oversette en relativ adresse bestemt under kompilering ved hjelp av prosessens base-adresse
4. Virtuelt minne brukes til å gi en illusjon av at vi har mer minne enn det vi fysisk har tilgjengelig
5. En sidetabell ("page table") er en tabell som oversetter mellom virtuelle og fysiske sider
6. En prosess sin sidetabell er alltid holdt i minnet på grunn av ytelse
7. Med en 32-bits adresse kan man adressere 4294967296 lokasjoner i minnet
8. Med en 32-bits adresse kan man adressere 1073741824 lokasjoner i minnet

#### 4b Nettverkslaget

Hvilke utsagn er riktige:

1. Nettverkslaget er øverste laget i en bro
2. IP (Internet protokollen) gir støtte for ruting og fragmentering
3. ARP (address resolution protocol) brukes for å finne MAC-adressen til en mottaker når IP-adressen er kjent i et LAN
4. IP er en forbindelsesorientert protokoll
5. Data som sendes av IP vil som regel inneholde hodet generert av transportlaget
6. ARP bygger opp en rutingtabell med par av IP-adresser og transportlagsporter slik at man vet hvor man skal sende en pakke
7. IPv6 (IP versjon 6) er en utvidelse av IPv4 og legges over IPv4 i TCP/IP-stakken

(Fortsettes på side 5.)

---

```
1: int B(void)
2: {
3:     char *buf, *p, *q, r = 'w', s = 'r';
4:
5:     p = buf = (char *) malloc(6);
6:     strcpy(p, "hallo");
7:     *(p + 1) += 4;
8:     printf("%s ", p);
9:
10:    q = p + 1;
11:    p = &r;
12:
13:    printf("%c%c%c", *p, *(q + 3), s);
14:    printf("%c%c!\n", buf[3], (*(buf + 1) - 1));
15:
16:    free(buf);
17: }
```

---

Figur 2: Funksjon B

#### 4c C programmering

Anta at du har en annen funksjon med koden beskrevet i figur 2. Hvilke utsagn er riktige:

1. Tekststrengen **“hallo”** skrives ut på linje 8
2. Tredje parameter til `printf` på linje 14, `*(buf + 1) - 1`, betyr at vi peker på andre byte i minneområdet allokert på linje 5, og vi trekker 1 fra verdien til andre byte
3. Tekststrengen **“ld!”** og et linjeskift (“newline”) skrives ut på linje 14
4. Funksjonen vil ikke kompilere
5. Funksjonen skriver ut tekststrengen **“hello world”**