

**Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap**



EKSAMENSOPPGAVE I FAG TDT4186 – OPERATIVSYSTEMER

Faglig kontakt under eksamen: Orestis Gkorgkas

Tlf.: 98832238

Eksamensdato: 11. august 2012

Eksamenstid: 09.00-13.00

Tillatte hjelpemiddel: D: Ingen trykte eller håndskrevne hjelpemiddel tillatt. Bestemt, enkel kalkulator tillatt.

Språkform: Bokmål

Sensurdato: 1. september 2012

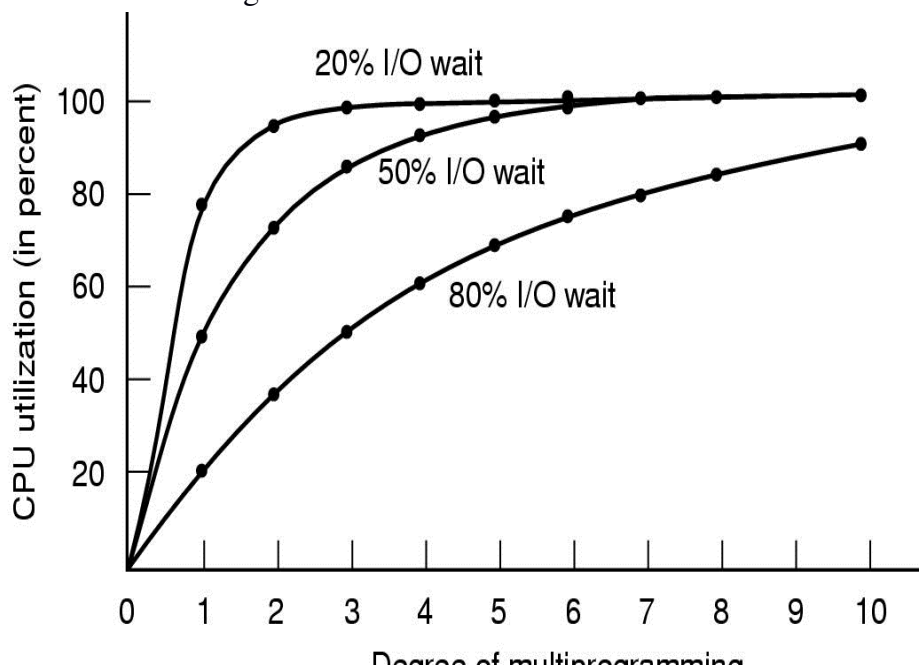
Oppgaven er kontrollert av professor Kjell Bratbergsengen.

Oppgave 1 – Generelt – 15 %

- Hvordan kan en brukerprosess få tak i operativsystemets tjenester, som for eksempel for å skrive data til en fil?
- Hva skjer når en prosess kjører en «TRAP»-instruksjon?
- Nevn noen systemkall som har å gjøre med håndtering av prosesser.

Oppgave 2 – Prosesser – 10 %

- Prosesser har tre hovedtilstander: *Running*, *Blocked* og *Ready*. Lag en figur som viser hvordan en prosess skifter mellom de tre tilstandene.
- Figuren under viser CPU-utnyttelsen som en funksjon av multiprogrammeringsgraden. Forklar kort hva figuren viser:



Oppgave 3 – Tråder – 10 %

Du har valget om å lage en serverprosess ved hjelp av en av de tre modellene: 1. tråder, 2. enkelttrådet prosess, 3. tilstandsmaskin (FSM). Hvilken modell vil du bruke og hvorfor? Forklar også hvorfor du ikke vil bruke de andre modellene.

Oppgave 4 – Synkronisering – 20 %

- Når forskjellige tråder skal bruke delte data, hvordan vil du hindre at trådene kommer i «race conditions», dvs. trådene ødelegger for hverandre? Diskuter litt de forskjellige løsningene du kjenner til.
- Programmet under gjør bruk av *wakeup* og *sleep*. *Producer* og *consumer* kjører i parallell, men har en fatal feil i synkroniseringen. Forklar hva som kan gå galt.

```

#define N 100                                     /* number of slots in the buffer */
int count = 0;                                   /* number of items in the buffer */

void producer(void)
{
    int item;

    while (TRUE) {                               /* repeat forever */
        item = produce_item();                  /* generate next item */
        if (count == N) sleep();                /* if buffer is full, go to sleep */
        insert_item(item);                      /* put item in buffer */
        count = count + 1;                      /* increment count of items in buffer */
        if (count == 1) wakeup(consumer);      /* was buffer empty? */
    }
}

void consumer(void)
{
    int item;

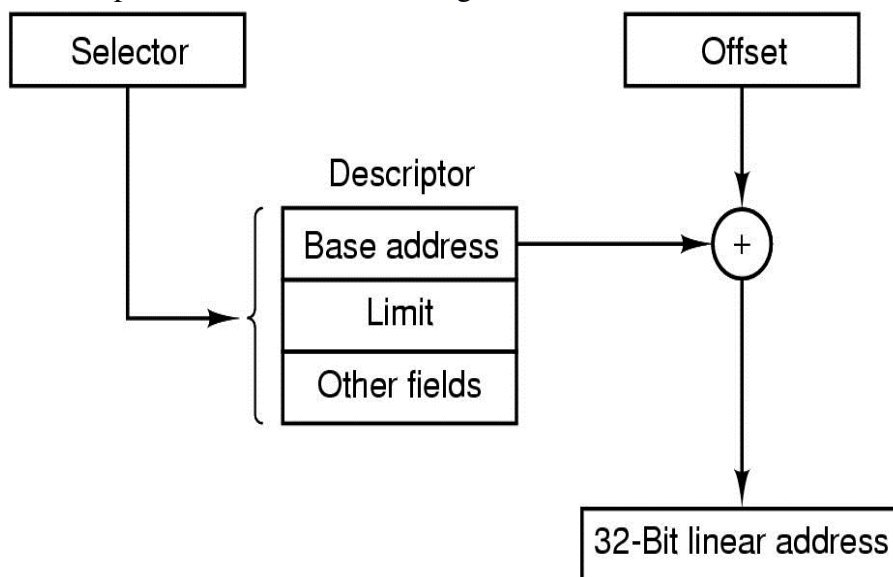
    while (TRUE) {                               /* repeat forever */
        if (count == 0) sleep();                /* if buffer is empty, got to sleep */
        item = remove_item();                   /* take item out of buffer */
        count = count - 1;                      /* decrement count of items in buffer */
        if (count == N - 1) wakeup(producer);  /* was buffer full? */
        consume_item(item);                     /* print item */
    }
}

```

Oppgave 5 – Minnehåndtering – 20 %

- a) Forklar hvordan man kan organisere en sidetabell slik at den kan støtte store adresserom. Bruk gjerne en figur.
- b) Forklar hvordan arbeidssettmodellen (working set) for sideerstatning virker.

- c) Under er det vist hvordan man regner ut adresser ved segmentering på Pentiumprosessoren. Forklar hva figuren viser:



Oppgave 6 – Filsystemer – 10 %

- Forklar hvordan man kan regne ut maksimal filstørrelse i et i-node-basert filsystem med støtte for enkelt-, dobbelt- og trippelt-indirekte blokker. Anta det i i-noden er 10 direktepekere og en peker av hver av de tre klassene med indirektepekere. Videre kan du anta at en blokk i filsystemet er 4K, dvs. 4096 bytes, og at en blokkpeker tar 4 bytes.
- Vi har et operativsystem som støtter «buffer cache» for filsystemet. Forklar hvordan vi kan få «buffer cachen» til å redusere sjansen for at filsystemet blir inkonsistent etter en krasj.

Oppgave 7 – Vranglås – 15 %

- Hvis et system har kommet i en vranglås, hvilke metoder finnes det for å få systemet ut av vranglåsen?
- Gitt et system med 3 prosesser (P0-P2) og fire typer ressurser (A, B, C, D). Vi har tre matriser som for hver ressurstype sier hvor mange enheter som er tilgjengelig, hvor mange enheter hver prosess har og hvor mange enheter hver prosess fremdeles spør etter:

Tilgjengelig:

A	B	C	D
2	1	0	0

Allokert:

	A	B	C	D
P0	0	0	1	0
P1	2	0	0	1
P2	0	1	2	0

Fremdeles spør etter:

	A	B	C	D
P0	2	0	0	1
P1	1	0	1	0
P2	2	1	0	0

Er systemet i en sikker tilstand og hvorfor/hvorfor ikke?