

EKSAMEN

Emnekode: ITF22506	Emne: Operativsystemer
Dato: 28. november 2013	Eksamenstid: kl. 9.00 til kl. 13.00
Hjelpemidler: Maks. tre A-4 ark med selvskrevede notater.	Faglærer: Jan Roland Olsson Einar Krogh Tlf. +46 703772914
Eksamensoppgaven: Oppgavesettet består av 4 sider inklusiv denne forsiden. Kontroller at oppgaven er komplett før du begynner å besvare spørsmålene. Oppgavesettet består av 6 oppgaver. Oppgavene 1, 2,3 er fra Modul 1 (JRO) Oppgavene 4, 5, 6 er fra Modul 2 (EK)	
Sensurdato: 20 Desember 2013 Karakterene er tilgjengelige for studenter på studentweb senest 2 dager etter oppgitt sensurfrist. Følg instruksjoner gitt på: http://www.hiof.no/index.php?ID=11229	

Oppgave 1 (17 %)

I denne oppgaven skal du selv lage BASH kommandoer og scripts.

- a) Skriv en kommando som flytter alle .java filer i nåværende katalog til katalogen /tmp.
- b) Anta at filen machines.txt inneholder IP-adresser til et antall datamaskiner. Lag en løkke som eksekverer kommandoen ping til hver IP-adresse i filen.
- c) Skriv et BASH script som tar to tall som argumenter på kommandolinjen og siden skriver ut det minste tallet.

Oppgave 2 (17 %)

Forklar så nøye du kan alt i følgende C program og hva som skjer når det eksekveres.

```
#include <stdio.h>

void f( char *fileName ) {
    FILE *F;
    int c;
    F = fopen( fileName, "r" );
    while(1) {
        c = getc( F );
        if( c == EOF )
            break;
        putc( c, stdout );
    }
    fclose( F );
}

int main( int argc, char* argv[] ) {
    int ctr;

    for( ctr = 1; ctr < argc; ctr = ctr + 1 )
        f( argv[ctr] );
    return 0;
}
```

Oppgave 3 (17 %)

Forklar kort hva som hender ved eksekvering av følgende Linux kommandoer.

- a) `grep print *.c`
- b) `cp /local/*?? /tmp`
- c) `ls -R / | grep test`
- d) `files=`ls -l | grep rw----- | wc -l``

Oppgave 4 (17 %)

- a) Hva menes med prosessplanlegging (Processor Scheduling)? Beskriv noen ulike algoritmer for prosessplanlegging. Hvordan foregår prosessplanlegging i Windows?
- b) Hvordan kan man på best mulig måte utnytte at en datamaskin har flere prosessorer. Nevn noen måter som organisering av bruk av flere prosessorer har vært forsøkt på. Hva menes med Symmetric Multi Processing (SMP)?
- c) Gi en definisjon av følgende begreper: Tråd, trådustifting, Spinlock, Mutex, kritisk region, Quantum, og round robin.

Oppgave 5 (17 %)

- a) To systemmekanismer i Windows operativsystem er avbruddsignaler (Interrupts) og unntakssignaler (Exceptions). Beskriv oppgaven til disse to systemmekanismene.
- b) Hvordan foregår kommunikasjon mellom maskinvare (Hardware) og brukerprogrammer (Applications) på en datamaskin? Hva gjør kommunikasjon mellom brukerprogrammer og maskinvare til et problem? Hva er oppgaven til Hardwarelaget (HAL)?
- c) Nevn noen årsaker til at Windows operativsystem kan krasje. Hva vil Windows gjøre om det oppstår en kritisk feil under programkjøring.

Oppgave 6 (17 %)

- a) Gi en definisjon av følgende begreper: namespace, managed pointer, delegate, project, solution, gcnew, CLR og class ArrayList.
- b) Du skal skrive koden til en klasse med en tråd i C++ (Console program i Visual Studio). Tråden skal skrive tallene 1,2,3, ..., 1000 ut på skjermen med 2 sekunders mellomrom. Fargen på tallene skal skifte for hvert tall. Når tråden har kommet til 1000 begynner den på ny. Bruk en Mutex for å synkronisere utskriften til skjermen.
- c) Skriv en funksjon i klassen under punkt b som heter StartTråd(). Denne funksjonen skal kunne brukes til å starte en eller flere nye tråder.

Vedlegg til oppgave 6

Thread Members

Thread(ThreadStart)	Initializes a new instance of the <u>Thread</u> class, specifying a delegate that allows an object to be passed to the thread when the thread is started.
Start()	Causes a thread to be scheduled for execution.
Sleep(Int32)	Blocks the current thread for the specified number of milliseconds.

ThreadStart Delegate

ThreadStart()	When a managed thread is created, the method that executes on the thread is represented by a ThreadStart delegate that is passed to the <u>Thread</u> constructor.
---------------	--

Mutex Members

Mutex()	Initializes a new instance of the Mutex class with default properties.
<u>WaitOne()</u>	Blocks the current thread until the current <u>WaitHandle</u> receives a signal.
ReleaseMutex()	Releases the Mutex once.