NTNU
Det skapande universitetet

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4258 Energieffektive datamaskinsystemer

Faglig kontakt under eksamen: Asbjørn Djupdal

Tlf.: 909 39452

Eksamensdato: 29. mai 2013

Eksamenstid (fra-til): 09:00 - 12:00

Hjelpemiddelkode/Tillatte hjelpemidler: D / Ingen trykte eller håndskrevne hjelpemidler. Enkel godkjent kalkulator er tillat.

Annen informasjon:

Målform/språk: Bokmål

Antall sider: 5

Antall sider vedlegg: 0

Kontrollert av:

_____

Dato Sign

Oppgave 1 Flervalgsoppgave (16 poeng)

Du får 2 poeng for hvert riktig svar og 0 poeng hvis svaret mangler. Hvis svaret er feil gis -0.5 poeng. Kun ett alternativ er riktig.

a) Hva er riktig om prosessorsamlebånd?

1. Throughput (antall instruksjoner utført per tidsenhet) økes
2. Latency (tid det tar å utføre en enkelt instruksjon) minskes
3. Samlebånd forutsetter cache
4. Samlebånd forutsetter virtuelt minne

b) Hva brukes en interruptkontroller til?

1. Kontrollere at CPU håndterer interrupt innen deadline
2. Minke responstiden til interrupthandleren
3. Samle interruptlinjer fra flere I/O-enheter slik at mange enheter får mulighet til å gi CPU et interrupt
4. Implementere synkroniseringsmekanismer for interrupthandlere

c) Hvilken av følgende årsaker kan *ikke* føre til at en CPU får interrupt?

1. En bruker trykker på en knapp
2. En ALU-operasjon fører til overflyt
3. En I/O-enhet trenger mer data fra prosessoren
4. En DMA-enhet er ferdig med å overføre data

d) Hva er hensikten med symboltabellen i en objektfil?

1. Nødvendig for at kode i én objektfil skal kunne referere til andre objektfiler
2. Gjøre det mulig å ha private variabler
3. Gjøre disassemblering mer oversiktlig
4. Dokumentasjonsgenerering

e) Hva er den vanligste måten å få til preemptiv multitasking på?

1. Hver prosess gir eksplisitt fra seg kontrollen ved OS-kallet yield()
2. En egen HW enhet tar seg av prosesschedulering
3. Timerinterrupt som kjører prosesscheduler jevnlig
4. En egen systemservice i user space tar seg av schedulering

f) Hva er *ikke* riktig å si om drivere i Linux?

1. En bug i driveren kan ødelegge for hele systemet
2. En driver lages ofte som en kjernemodul
3. Grensesnittet til en driver er ofte gjennom device-filer
4. En driver har tilgang til alle vanlige C-bibliotek

g) Hva er *ikke* riktig om prosesser og tråder?

1. En prosess kan kommunisere med andre prosesser
2. En prosess har ofte sitt eget virtuelle adresserom
3. En tråd deler adresserom med andre tråder i samme prosess
4. En tråd deler som oftest adresserom med alle tråder i alle prosesser

h) Hva er *ikke* en vanlig OS-teknikk for å redusere energiforbruk til en datamaskin?

1. Justere spenning og klokkefrekvens avhengig av arbeidsbelastning
2. Endre sidetabeller for mer effektiv bruk av TLB
3. Skru av og på individuelle I/O-enheter dynamisk ut i fra bruksmønster
4. Sette CPU i sleep-mode når OS har vært idle en stund

Oppgave 2 Buss (12 poeng)

a) Du skal spesifisere en bussprotokoll for kommunikasjon mellom en CPU og to I/O-kontrollere som ligger på adressene 0x100 og 0x200.

Dette skal være en asynkron buss, hvor følgende signaler skal brukes:

- enable (1 bit)
- read/ write (1 bit)
- adresse (16 (bit)
- data (8 bit)
- enq (1 bit)
- ack (1 bit)

Tegn et timingdiagram hvor du viser hvordan du gjør følgende operasjoner:

1. Lese en byte fra adresse 0x100 (hvor tallet 170 ligger)
2. Skrive tallet 85 til adresse 0x200

Dersom du har gjort antagelser som ikke går klart fram i diagrammet må disse forklares med tekst.

b) Hva ligger i begrepet “burst transfer” for en synkron buss?

Oppgave 3 Kompilator og programmering (14 poeng)

Gitt følgende C-kode.

```
x = a + b
x = x + c
y = x + c
z = a - b
```

a) Omgjør C-koden til "single assignment form" og tegn tilhørende dataflytgraf (DFG). Navngi alle kantene i grafen med variabelnavn.

b) Gjør en levetidsanalyse for variablene.

Bruk graph-coloring til å finne antall registre som er nødvendig for å implementere dette i assembly.

c) Hva kan gjøres med programmet i punkt b som minsker antall nødvendige registre uten å endre sluttresultat?

d) Forklar hvordan en C-kompilator kan implementere funksjonskall som støtter rekursjon, argumenter, returverdi og lokale variabler.

Oppgave 4 Operativsystem (8 poeng)

Forklar begrepet "priority inversion" og beskriv hvordan dette kan håndteres av scheduleren.