Prosjektøving: Eksamenstrening

Denne prosjektøvingen skal gi trening på å skrive gode eksamensbesvarelser. Hver student skriver sin egen besvarelse. Deretter leveres det ut sensur-veiledning og noen eksempler på besvarelser på ulike karakternivå. Deretter skal hver student rette og gi tilbakemelding på besvarelsen til minst en medstudent. Tilslutt skriver studentene som har samarbeidet et notat som leveres på itslearning.

Eksamen: Datateknikk og operativsystem

Informasjon om eksamen:

Varighet/eksamenstid: 3 timer.

Hjelpemidler: Godkjent kalkulator emnegruppe 1.

Kandidaten må selv bringe med seg kalkulatoren.

Oppgave 1. Tallsystemer og datarepresentasjon (vekt: 15%)

I denne oppgaven brukes følgende to bitmønstrene:

Bitmønster A: 00010111

Bitmønster B: 10101110

1. Skriv hvert av de to bitmønstrene på både *heksadesimal* og *desimal* form. Vis fremgangsmåten nøye.

A: 00010111

Hexadesimal:

Deler opp tallet i 4 og 4 bit fra høyre

0001 0111

Regner om disse to binærtallene til desimal og får:

1. 7

**Altså: 1716**

Desimal:

Konverterer fra hexadesimal til desimal

Da får vi:

(7\*160)+ (1\*161)= 7 + 16 = **2310**

B: 10101110

Hexadesimal:

Deler opp tallet i 4 og 4 bit fra høyre

1010 1110

Regner om disse to binærtallene til hexadesimal og får:

A E

**Altså: AE16**

Desimal:

Hvert ettall tilsvarer 2n der n er posisjonen til ettallet fra høyre (starter på 0):

Altså får vi:

27 + 0 + 25 + 0 + 23 + 22 + 21 + 0 = 128 + 32 + 8 + 4 + 2 = **17410**

Oppgave 2. Busser (vekt: 15%)

De tre hovedtypene av busser som er omtalt i kurset er: *parallell buss*, *seriell buss* og *svitsjet* *buss*.

1. Beskriv oppbyggingen til hver av disse hovedtypene.

**Parallell buss:**

Består av mange ledninger/kobberbaner som ligger tett inntil hverandre i samme retning og gjør at man kan sende flere forskjellige beskjeder med bussen samtidig. Hver kobberbane kan imidlertid kun være enten av eller på. Si at du vil sende et 16bit signal, da trenger du 16 forskjellige kobberbaner, 1 for hver bit for å sende signalet samtidig.

**Seriell buss:**

Består ev en enkelt kobberbane/ledning og vil derfor kun kunne levere ett og ett signal om gangen, men kan derimot ha flere signaler «reisende» gjennom kobberbanen samtidig. Altså vil du kunne sende de samme 16 bit fra eksempelet over i den samme kobberbanen separert etter tid for å skille hver bit fra hverandre.

**Svitsjet buss:**

Oppgaven til denne bussen er å koble sammen riktige komponenter og la dem kommunisere uforstyrret samtidig som andre komponenter. En svitsjet buss fungere som et knutepunkt hvor flere forskjellige komponenter kan være koblet. Svitsjen vil så ha flere «lanes» gjerne, en til og fra hver komponent.

1. Nevn fordeler og ulemper med hver av disse hovedtypene.

**Parallell buss:**

* **Fordeler:**
  + Rask overføring når ledningene er korte nok.
  + Kan sende en større mengde data på en gang.
* **Ulemper:**
  + Er avhengig av korte avstander for å ha god overføringskapasitet og hastighet

**Seriell buss:**

* **Fordeler:**
  + Krever relativt liten plass, trenger kun en ledning
  + Med høy frekvens er serielle busser mer praktisk ved større avstander da parallelle buser er svært avhengig av avstand for å øke hastigheten.
* **Ulemper:**
  + Krever høy frekvens for at den skal være rask, og er avhengig av en kontroll på hver ende for å holde styr på bitsene som blir sendt.

**Svitsjet buss:**

* **Fordeler:**
  + Flere komponenter kan snakke med hverandre samtidig.
* **Ulemper:**
  + Teknisk komplisert og krever at alle tilkoblede komponenter er kompatible med den svitsjede bussen

1. Moderne datamaskiner bruker gjerne alle disse hovedtypene av busser. Nevn minst et eksempel på moderne busser fra hver av hovedtypene.

**Parallell buss:**

* SCSI

**Seriell buss:**

* USB
* SATA

**Svitsjet buss:**

* PCI-express

Oppgave 3. Cache og systemarkitektur (vekt: 20%)

1. Hva sier prinsippet om lokalitet? Begrunn hvorfor prinsippet gjelder.

Lokalitetsprinsippet sier at en prosess svært ofte har instuksjoner som ligger enten sekvensielt, eller i nærheten av hverandre, dette gjør at en cache med god sannsynlighet vet hvor omtrent hvor neste aksess skje.

1. L1-cachen til en prosessor oppgis til å være 8-veis sett-assosiativ cache med linjestørrelse på 64 byte. Forklar nøye begrepene som er understreket, og hvorfor det ofte er en hensiktsmessig oppbygging.

**8-veis sett-assosiativ:**

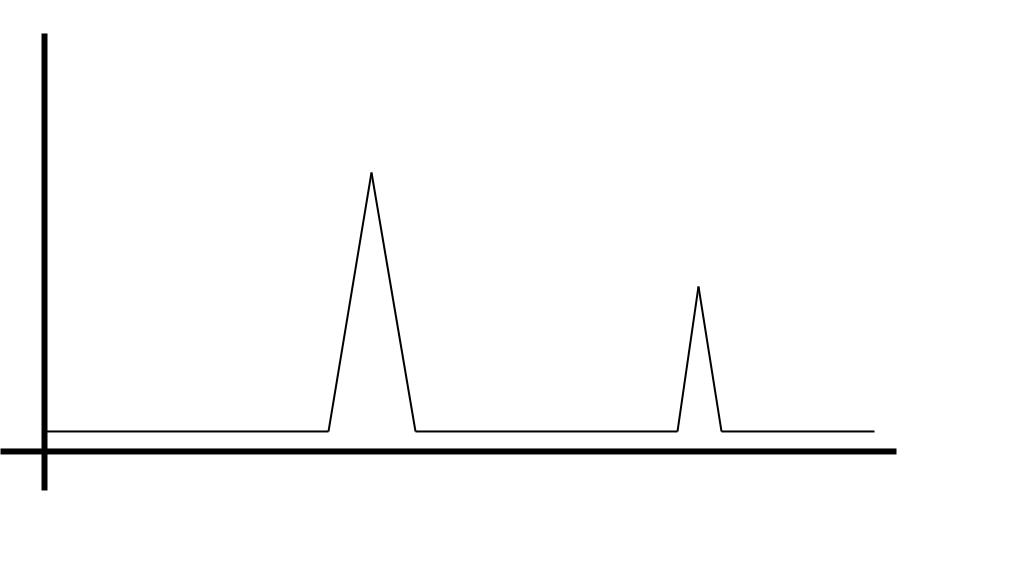
Betyr at cachen har 8 mulige «blokker» med plass til en linje den henter ut fra minnet.

**Linjestørrelse:**

Forteller hvor stor kapasitet hver «linje» cachen henter ut har. Altså hvor mye cachen henter ut fra minnet når den ikke finner riktig minnelokasjon i cachen sin. At linjestørrelsen er 64byte betyr altså at cachen vil hente 64byte fra minnet hver gang den er nødt til å gå ut i minnet for å hente informasjon.

1. Dersom man tar et normalt program og ser hvilke minnelokasjoner som i praksis brukes av programmet, vil man se at det bruker to områder i minnet. Dette er skjematisk fremstilt i figuren nedenfor. Plasseringen av disse områdene i minnet vil bestemmes av operativ-systemet.

Sannsynlighet for neste aksess



Minneadresse

Svar på følgende spørsmål:

* + Hvorfor er det *to* slike områder?

Data og intruskjoner ligger i to forskjellige steder i minnet, sannsynliugheten for at neste aksess er for å hente ut en av disse er relativt stor, og siden man vet omtrent hvilken minnelokasjon denne aksessen vil skje i får vi disse to områdene.

* + Har denne egenskapen med programmene noe å si for at sett-assosiativ cache er den vanligste mapping-funksjonen?

En sett assosiativ cache er svært hensiktsmessig her da den har muligheten for å ha flere linjer i cachen samtidig, altså kan f.eks. 4 blokker være instruksjoner og 4 blokker være data (fra n 8-veis sett assosiative cachen fra tidligre oppgave). På denne måten slipper cachen og hele tiden bytte ut instruksjonen for data, og data for intruksjoner om hverandre.

1. Moderne minneteknologier er spesiallaget for å utføre såkalte «burst».
   * Hva er en «burst»?

Em burst er en kontinuerlig

En burst hender når cahcen er hentet inn instruksjoner fra minnet. Da vil prosessoren bli gitt en lokasjon før den deretter utfører mange instruksjoner sekvensielt etter hverandre.

* + - * + - Med utgangspunkt i det vi har lært om systemarkitektur: Forklar detaljert hvordan en «burst» utføres.

Ehhhhhh cachj?

Oppgave 4. Prosesser og tråder (vekt: 25%)

1. Vi snakker ofte om både prosesser og tråder. Forklar hva en prosess er og hva en tråd er og hvilken sammenheng det er mellom prosesser og tråder.

En prosess er en kjørende del av et program, en tråd er

1. Hvordan er det mulig å kjøre flere tråder på en CPU enn det er tilgjengelige kjerner i CPU-en? Forklar.

Dette er mulig på grunn av det som kalles Thread Control Block (TCB) som håndtere alle de forskjellige trådene, hver tråd har sin egen TCB som holder styr på hvilken modus tråden er i, når tråden skal utføres (i forhold til andre tråder) osv. TCB sender så informasjonen til kjernen i den oppsatte rekkefølgen. Hver tråd trenger heller ikke fullføres før det bytter til en annen tråd, da en tråd kan bli «kastet ut» av en tråd med høyere prioritet eller andre årsaker.

1. Tjenerprogrammer, f.eks på en filtjener eller på en webtjener, er som oftest såkalt ”multithreaded”, dvs kjører mange tråder. Hvorfor er det hensiktsmessig å kjøre mange tråder på f.eks en filtjener sammenlignet med å kjøre kun en prosess? Forklar.

Det er svært hensiktsmessig å kjøre mange tråder samtidig ettersom dette gjør det mulig å utføre flere forskjellige prosesser samtidig. En webtjener bør for eksempel ha muligheten til å håndtere bruk fra flere forskjellige brukere samtidig.

1. Prosesser får tilgang til i/o-enheter via operativsystemkjernen. På hvilken måte? Forklar.

For at prosesser skal få tilgang til i/o enheter er de nødt til å spørre operativsystemkjernen om dette. Dette gjøres med såkalte forhåndsbestemte «System calls». Får operativsystemkjernen en System-Call går kjernen over til privilegert modus og kan bruke den forspurte i/o enheten.

1. Hvorfor kan ikke prosessene gå direkte mot i/o-enhetene og gjøre sin egen i/o uavhengig av operativsystemkjernen? Forklar.

Prosesser blir forhindret fra å gå direkte mot i/o-enheter ettersom dette utgjør en stor sikkerhetsrisiko, både med tanke på informasjon og personlig sikkerhet. Dersom et program kunne fritt aktivert og brukt i/o-utstyr åpner det opp for problemer som å laste over filer fra en ekstern harddisk uten samtykke, skru på kamera/mikrofon, bruke skjermen osv.

Oppgave 5. Minne (vekt: 25%)

1. I minneadministrasjon snakker man om relokering av adresser som betyr omregning fra virtuelle adresser til fysiske adresser. Forklar hvorfor man trenger denne omregningen fra virtuelle til fysiske adresser. Forklar også hvordan denne omregningen foregår i dagens operativsystemer som f.eks i Windows eller Linux (ikke tekniske detaljer her, kun prinsipper om hvordan det foregår).

Omregning fra virtuelle til fysiske adresser

1. Du har en datamaskin med 64 bits CPU. Fysisk minne er på 16 Gbyte. Men hvor stort er det virtuelle minnet? Jeg er ikke ute etter et eksakt tall her, men en diskusjon rundt temaet, f.eks vil jeg vite noe om både hvor stort virtuelt minne operativsystemet kan sette av og hvor mye virtuelt minne hver prosess disponerer.

Det virtuelle minnet virker som proseessene til å være uendelig.

3