

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk

Institutt for datateknikk
og informasjonsvitenskap

Løsning på kontinuasjon i TDT4186 Operativsystemer
August 2005, 0900-1300

Det ønskes korte og konsise svar på hver av oppgavene. Det vesentlige er å kunne dokumentere forståelse, beherske prinsipper og se sammenhenger - ikke å kunne gjengi en mengde detaljer.

Der det synes å mangle noen opplysninger, må det angis hvilke antagelser som synes å være naturlige. Merk at viktige begreper er angitt på både norsk og engelsk.

Oppgave 1: Bruk av CPU (CPU Management) – 24 %

- a) Forklar hvordan prosesser (PROCESSES) representeres i en datamaskin!

SVAR:

- Prosesser representeres med en datastruktur som inneholder områder for program, data, stakk og kontekst. Denne datastrukturen kan lenkes inn og ut av ulike køer i.f.m. ulik håndtering av ressurser.

- b) Diskuter fordeler og ulemper med tråder (THREADS) kontra prosesser (PROCESSES).

SVAR:

- Prosesser passer best for mer isolerte aktiviteter, mens tråder passer best for mer samhørende aktiviteter. Prosesskifter er dyrere enn trådsifter.

- c) Hvorfor er bruk av henholdsvis mikrokjerne (MICRO KERNEL) og objektorientering (OBJECT ORIENTATION) fornuftige måter å organisere operativsystemer på?

SVAR:

- Mikrokjerner passer godt fordi minimaliseringen av programkode som kjører i kjernemodus gir økt sikkerhet. Objektorientering fungerer godt fordi oppsplittingen i flere, uavhengige moduler tilfredsstiller behovene til distribuerte systemer.

Oppgave 2: Kontroll av prosesser (Process Synchronization) – 16 %

- a) Forklar hvordan prosesser (PROCESSES) styres slik at de ikke ødelegger for hverandre!

SVAR:

- Prosesser kontrolleres med semaforer / monitorer / meldinger slik at kun en prosess er inni gjensidig avhengige kritiske regioner om gangen, og slik at deres tilhørende aktiviteter ordnes i tid slik at tilsvarende overordnede krav overholdes.

- b) Diskuter fordeler og ulemper med programvarebasert (SOFTWARE BASED) prosessstyring kontra maskinvarebasert (HARDWARE BASED) prosessstyring.

SVAR:

- Maskinvarebasert styring må brukes på laveste nivå – dvs. der det er meget korte kritiske regioner som innvolveres, mens programvarebasert styring må brukes der det er mer lange kritiske regioner som innvolveres – dvs. på høyere nivå.

Oppgave 3: Bruk av lager (Memory Management) – 20 %

- a) Forklar hvordan lokalitetsprinsippet (LOCALITY PRINCIPLE) anvendes til å effektivisere lagerbruk!

SVAR:

- Lokalitetsprinsippet tilsier at kode / data romlig nær kode / data som aksesseres akkurat nå, gjerne aksesseres igjen innen kort tid. Således lastes og lagres mer kode / data til primærlageret enn det som akkurat trengs ved aksess til sekundærlageret.

- b) Diskuter fordeler og ulemper med "Minst nylig referert" (LRU)-algoritmen kontra "4 sjansers klokke" (UM-CLOCK)-algoritmen.

SVAR:

- UM-CLOCK har mindre overhead med brukbar ytelse, mens LRU har bedre ytelse med stor overhead.

Oppgave 4: Kjøring av prosesser (Process Scheduling) – 16 %

- a) Forklar hvordan prosessavbryting (PROCESS PREEMPTION) anvendes til å effektivisere prosesskjøring!

SVAR:

- Prosessavbryting tillater at lange prosesser kan fratas ressurser før de er ferdige. Korte prosesser kan således komme raskere til de ressursene de behøver. Dette senker den gjennomsnittlige utførelsestid.

- b) Diskuter fordeler og ulemper med Lastdelings-algoritmen (LOAD SHARING) kontra Samkjørings-algoritmen (GANG SCHEDULING).

SVAR:

- GANG SCHEDULING sikrer en bedre prosessytelse mot en mindre ressursutnyttelse ved at trådene sikres å bli kjørt sammen, mens LOAD SHARING gir en bedre ressursutnyttelse mot en mindre prosessytelse ved at trådene driver selvtilordning av CPUer.

Oppgave 5: Bruk av I/O (I/O Management) – 24 %

- a) Forklar hvordan ulike diskallokering (DISK ALLOCATION)-metoder anvendes til å effektivisere I/O-bruk!

SVAR:

- Allokering med sammenhengende enheter kan gi lav aksestid for en enkelt prosess. Mens allokering av kjedete enheter kan gi lav plassoverhead for alle prosesser samlet. Og allokering av indekserte enheter kan gi en balanse mellom aksestid og plassoverhead.

b) Diskuter fordeler og ulemper med "Korteste søk først" (SSTF)-algoritmen kontra "Enveis heis" (C-SCAN)-algoritmen.

SVAR:

- C-SCAN gir en lav varians i responstid i.f.m. diskaksesser, mens SSTF gir et lavere snitt på responstid i.f.m. diskaksesser.

c) Hvorfor er en "Frekvensbasert stakk" (FREQUENCY BASED STACK)-mekanisme en fornuftig måte å cache I/O-data på?

SVAR:

- Denne mekanismen tilsvarer en kombinasjon av LRU og LFU mekanismene som passer den støtvide forekomsten som er typisk for I/O data. For caching av I/O data kan man også ta den ekstra overheaden som disse mekanismene tilsier.