

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk

Institutt for datateknikk
og informasjonsvitenskap

Kontaktperson under eksamen:

Institutt for datateknikk og informasjonsvitenskap, Gløshaugen

Mads Nygård, 93440.

Eksamen i TDT4186 Operativsystemer (Bokmålstekst)

Tirsdag 14. desember 2004, kl. 09.00-13.00

Hjelpemidler: D.

Ingen trykte eller håndskrevne hjelpemidler tillatt.

Bestemt, enkel kalkulator tillatt.

Sensuren faller 19. januar.

Det ønskes korte og konsise svar på hver av oppgavene.

Les oppgaveteksten meget nøye, og vurder hva det spørres etter i hver enkelt oppgave.

Dersom du mener at opplysninger mangler i oppgaveformuleringene, beskriv de antagelsene du gjør.

Oppgave 1: Bruk av CPU og I/O (CPU & I/O Management) – 20 %

- a) Forklar i hvilke tilfeller en bør bruke henholdsvis prosesser (Processes) og tråder (Threads).
- b) Diskuter hvordan modusskifter (Mode Switching) utføres.
- c) Diskuter hvordan I/O-bufring (I/O Buffering) utføres.
- d) Angi detaljert - med formell kode eller uformell beskrivelse - hvordan "Frekvensbasert stakk" (Frequency Based Stack)-metoden for I/O-caching virker.

Oppgave 2: Kontroll av prosesser (Process Synchronization) – 20 %

- a) Forklar i hvilke tilfeller en bør bruke henholdsvis programvarebasert (Software Based) prosessstyring og maskinvarebasert (Hardware Based) prosessstyring.
- b) Diskuter hvordan prosesser (Processes) kan kommunisere trygt og effektivt med hverandre.
- c) Angi detaljert - med formell kode eller uformell beskrivelse - hvordan "Oppdage og opprette" (Deadlock Detection)-metoden for vranglåshåndtering virker.

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

- a) Forklar i hvilke tilfeller en bør bruke henholdsvis fast partisjonering (Fixed Partitioning), enkel sidedeling (Simple Paging) og virtuelt lager med sidedeling (Virtual Memory with Paging).
- b) Diskuter hvordan side- og segmenttabeller (Page & Segment Tables) kan struktureres slik at de kan aksesseres effektivt og riktig.
- c) Angi detaljert - med formell kode eller uformell beskrivelse - hvordan "4 sjansers klokke" (UM-Clock)-algoritmen virker.

Oppgave 4: Kjøring av prosesser (Process Scheduling) – 20 %

- a) Forklar i hvilke tilfeller en bør bruke henholdsvis prosessavbryting (Process Preemption) og prosessprioritering (Process Prioritization).
- b) Diskuter hvordan prosesser og tråder kan tidsstyres annerledes på multiprosessorer (Multi CPUs) enn på singelprosessorer (Single CPUs).
- c) Angi detaljert - med formell kode eller uformell beskrivelse - hvordan "Høyeste responsforhold først" (Highest Response Ratio Next)-algoritmen virker.

Oppgave 5: Distribuerte systemer (Distributed Systems) – 20 %

- a) Forklar i hvilke tilfeller en bør bruke henholdsvis multiprosessorer (Multi CPUs) og distribuerte systemer (Distributed Systems).
- b) Diskuter hovedprinsippene i gruppekommunikasjon (Group Communication).
- c) Diskuter hovedkomponentene i CORBA (Common Object Request Broker Architecture).
- d) Angi detaljert - med formell kode eller uformell beskrivelse - hvordan den helt distribuerte algoritmen (altså ikke ringalgoritmen) for gjensidig utelukkelse (Mutual Exclusion) virker.