3.4 Process control

Tuesday, April 7, 2020 1:07 AM

De fleste prosessorer støtter to moduser av execution. Noen instruksjoner kan bare executes i en mer-priviligert modus. Disse inkluderer lesing og endring av et kontrollregister, som PSW, primitive I/O-instruksjoner, og instruksjoner som relaterer til minnehåndtering. I tillegg er det noen deler av minne som bare kan aksesseres i den mer-priviligerte modusen. Den mer-priviligerte kalles gjerne systemmodus, kontrollmodus eller kjernemodus (*kernel mode*). Denne modusen referer til kjernen av OS-et, som kontrollerer de mer viktige systemfunksjonene. Den mindre-priviligerte modusen kalles brukermodus. Tabellen under viser de typiske funksjonene i kjernemodus:

Table 3.7 Typical Functions of an Operating System Kernel

Process Management
• Process creation and termination
• Process scheduling and dispatching
• Process switching
• Process synchronization and support for interprocess communication
• Management of process control blocks
Memory Management
• Allocation of address space to processes
• Swapping
• Page and segment management
I/O Management
• Buffer management
• Allocation of I/O channels and devices to processes
Support Functions
• Interrupt handling
• Accounting
• Monitoring

Det er nødvendig å adskille kjernemodusen fra brukermodus for å beskytte OS-et og systemtabellene, som for eksempel prosesskontrollblokker fra å bli tuklet med av brukerprogrammer. I kjernemodus har softwaren fullstendig kontroll på prosessoren og alle dens instruksjoner, registre og minne. Prosessoren vet hvilken modus den kjører ved hjelp av en bit i PSW.

Prosessoppretting

Som nevnt tidligere er det fire grunner til at en prosess opprettes. Når er prosess opprettes gjøres det på følgende måte av OS-et:

1. Tildeler prosessen en unik PID (prosess-ID). På dette tidspunktet legges det til en ny rad til hovedprosesstabellen.
2. Tildeler minne til prosessen. Dette inkluderer alle elementer av prosessbildet. OS-et må altså vite hvor mye plass som er nødvendig for den private brukers adresserom (programmer og data) samt brukerstacken. TIl slutt må OS-et tildele rom for prosesskontrollblokken.
3. Initialiserer prosesskontrollblokken. Tilstandsinformasjonen til prosessen setters som regel til 0 på de fleste innganger (bits) utenom programtelleren som er satt til programmets entry point.
4. Setter passende linker. For eksempel hvis OS-et håndterer hver planleggingskø som en lenket-liste, så må være prosess bli satt til Ready eller Ready/Suspend-listen.
5. Opprettelse eller utvidelse av andre datastrukturer.

Prosess-skifte (*process switching)*

Et prosess-skifte kan oppstå når som helst OS-et har gjenopptatt kontroll fra den kjørende prosessen. Dette skjer gjerne i forbindelse med avbrudd, og det finnes to typer systemavbrudd; vanlige avbrudd, og såkalte feller (*trap*). Den første typen avbrudd skjer som en følge av en hendelse eksternt eller uavhengig av den kjørende prosessen, som eksempelvis en I/O-operasjon. Feller relaterer en error eller en unntaksbetingelse generert av den kjørende prosessen (e.g. ulovlig forsøk på filaksess). Med et vanlig avbrudd blir kontrollen først satt til avbruddshåndtereren, som først gjør sine greier og så går kontrollen tilbake til en OS-rutine som håndterer nøyaktig den typen avbrudd som oppstod. Eksempler er:

- Klokkeavbrudd: OS-et sjekker om den nåværende kjørende prosessen har executet over den maksimale tidsgrensen, referert til som *time slice*. Time slice er den maksimale tidsbruken en prosess kan execute før den blir avbrutt. Hvis dette skjer så må den endres til Ready tilstand og en annen prosess må dispatche.
- I/O-avbrudd: OS-et avgjør hvilken type I/O-operasjon som har oppstått. Hvis I/O-operasjonen gjør at flere prosesser må vente så settes alle de korresponderende blokkerte prosessene til Ready-tilstand. Deretter må OS-et avgjøre om prosessen som akkurat kjører skal fortsette eller gi plass til en høyere-prioritert prosess i Ready.
- Minnefeil: Prosessoren kan lese av en virtuell minneadresse som referer til et ord som ikke er i hovedminne. I så fall må OS-et hente inn blokken (*page* eller *segment*) av minne som inneholder referansen fra sekundærminnet til hovedminnet. Etter I/O-forespørselen om sending av blokken til hovedminnet, så må prosessen med minnefeilen settes til Blocked. OS-et utfører så et skifte for å fortsette en annen prosess. Etter blokken er hentet settes prosessen med minnefeilen til Ready, da den ikke har en minnefeil lengre.

Med en såkalt *felle*, så avgjør OS-et om erroren eller unntaksbetingelsen er fatal. Hvis ja, så settes den nåværende kjørende prosessen til Exit for å gi plass til et prosess-skifte. Hvis ikke, så må OS-et håndtere etter samsvar med alvorlighetsgraden og typen av error. OS-et kan også aktiveres av et overvåkningskall (*supervisor call*) fra programmet som executes.

Table 3.7 Typical Functions of an Operating System Kernel

Process Management
• Process creation and termination
• Process scheduling and dispatching
• Process switching
• Process synchronization and support for interprocess communication
• Management of process control blocks
Memory Management
• Allocation of address space to processes
• Swapping
• Page and segment management
I/O Management
• Buffer management
• Allocation of I/O channels and devices to processes
Support Functions
• Interrupt handling
• Accounting
• Monitoring

Modusskifte

I kap 1 så vi på instruksjonssyklusen der det var et avbruddsstadiet på slutten av syklusen. I dette stadiet sjekker prosessoren om det er noen aventende avbrudd, indikert ved avbruddssignal. Hvis det ikke er noen aventende avbrudd fortsetter prosessoren til fetch-stadiet og fetcher neste instruksjon for det nåværende programmet i den nåværende prosessen. Dersom en et avbrudd er aventende gjør prosessoren følgende:

1. Den setter programtelleren til startadressen til avbruddshåndterer-programmet.
2. Den skifter fra brukermodus til kjernemodus så avbrudd-prosesseringskoden kan inkludere priviligerte instruksjoner.

Prosseroen fortsetter så til fetch-stadiet og fetcher den første instruksjonen i avbruddshåndterer-programmet, som tjenestegjør avbruddet. På dette tidspunktet er konteksten til prosessen som ble avbrutt lagret i prosesskontrollblokken til det avbrutte programmet. Informasjonen som lagres il prosesskontrollblokken inneholder den nødvendige informasjonen til at prosessen som ble avbrutt kan gjenopptas og fortsette execution.

Endring av prosesstilstand

Det er altså en klar forskjell mellom et modusskifte og prosess-skifte. Et modusskifte kan skje uten at tilstanden til prosessen som kjører blir endret. Dersom den nåværende kjørende prosessen derimot skal endres til en annen tilstand, så må OS-et gjøre flere endringer i sitt miljø. Disse stegene involverer et fullstendig prosess-skifte:

1. Lagre konteksten til prosessoren, samt programtelleren og andre registre.
2. Oppdatere prosesskontrollblokken til prosessen som er i nåværende kjørende tilstand. Andre relevante felter må også oppdateres, inkludert grunnen til hvorfor den ble endret fra kjørende tilstand og bokføringsinformasjon.
3. Flytte prosesskontrollblokken av denne prosessen til sin passende kø.
4. Selekter en annen prosess for execution.
5. Oppdatere prosesskontrollblokken til prosessen som ble selektert. Dette inkluderer å endre tilstanden til denne prosessen til kjørende.
6. Oppdatere minnehåndterings-datastrukturer. Dette kan være påkrevd, avhengig av hvordan adresseoversetting er håndtert.
7. Gjenopprette konteksten til prosessoren som eksisterte ved tidspunktet den selekterte prosessen sist ble skiftet til kjørende tilstand, ved å laste inn de tidligere verdiene fra programtelleren og andre registre.

Altså krever et prosess-skifte, som involverer en tilstandsendring, mer innsats enn et modusskifte.