

3.1 What is a process?

Sunday, April 5, 2020 3:39 PM

Før vi ser nærmere på begrepet *prosess* kan vi oppsummere noen av konseptene fra kap 1 & 2.

1. En computerplattform består av en samling av hardwareressurser, som prosessor, hovedminne, I/O-moduler, timers, harddisker osv.
2. Computerapplikasjoner er utviklet til å utføre en oppgave. De aksepterer typisk et input, etterfulgt av prosessering, og genererer så et output.
3. Det er ineffektivt å skrive applikasjoner til en bestemt hardwareplattform. Grunnene til dette er:
 - a. Flere applikasjoner kan utvikles på samme plattform. Derfor gir det mening å utvikle en generell rutine for å aksessere computerens ressurser.
 - b. Prosessoren tilbyr selv begrenset støtte for multiprogrammering. Software er derfor nødvendig for å håndtere delingen av prosessoren og andre ressurser av flere applikasjoner samtidig.
 - c. Når flere applikasjoner er aktive samtidig, er det nødvendig å beskytte dataen, I/O-bruken, og andre ressurserbruk av hver applikasjon fra de andre.
4. OS-et ble utviklet for å tilby et bekvemmelig, funksjonsrikt, sikkert og konsistent grensesnitt for applikasjonene i bruk. OS-et er et lag av software mellom applikasjoner og computer hardware som støtter applikasjoner og hjelpemidler.
5. Vi kan anse OS-et å tilby en uniform, abstrakt representasjon av ressurser som kan forespørres og aksesseres av applikasjoner. Ressurser inkluderer hovedminne, nettverksgrensesnitt, filsystemer osv. Når OS-et har opprettet disse ressursabstraksjonene for applikasjonen å bruke, så må den også håndteres deres bruk. For eksempel kan et OS tillate ressursdeling og ressursbeskyttelse.

Disse konseptene av applikasjoner, systemsoftware og ressurser gjør at vi kan se nærmere på hvordan et OS kan håndtere applikasjoner slik at:

- Ressurser kan gjøres tilgjengelig til flere applikasjoner.
- Den fysiske prosessor kan byttes mellom flere applikasjoner så alle vil fremstå å progrediere.
- Prosessoren og I/O-enheter kan brukes effektivt.

Alle moderne operativsystemer avhenger av en modell hvor execution av applikasjoner korresponderer til eksistensen av en eller flere prosesser.

Prosesser

Fra kap 2 hadde vi at en definisjon på en prosess sammenfaller med:

- Et program i execution
- Et instans av et program som kjører på en computer
- En entitet som kan tildeles til og executes på en prosessor
- En enhet av aktivitet karakterisert av execution av en sekvens med instruksjoner, en nåværende tilstand, og et assosiert sett med systemressurser.

Vi kan også tenke på en prosess som en entitet som inneholder et antall elementer. To essensielle elementer av en prosess er *programkoden* (som kan deles mellom andre prosesser som executer samme program) og et *datasett* tilhørende programkoden. Når prosessoren starter å execute denne programkoden, omtaler vi denne utførende enheten som en prosess. På ethvert tidspunkt, mens programmet executes, kan denne prosessen unikt karakteriseres ut fra flere elementer:

- **Identifikator (identifiser):** En unik identifikator assosiert med denne prosessen, for å skille den fra andre prosesser.
- **Tilstand (state):** Hvis prosessen holder på å executes, er den i en kjørende tilstand (running state).
- **Prioritet (priority):** Prioritetsnivå relativt til andre prosesser.
- **Programteller (PC):** Adressen til den neste instruksjonen i programmet som skal executes.
- **Minnepekere:** Inkluderer pekere til programkoden og dataen assosiert med denne prosessen, i tillegg til minneblokker delt med andre prosesser.
- **Kontekstdata:** Dette er data som er tilstede i registerene i prosessoren mens prosessen blir executed.
- **I/O-statusinformasjon:** Inkluderer utestående I/O-forespørslar, I/O-enheter tildelt denne prosessen, en liste av filer i bruk av prosessen osv.
- **Bokføringsinformasjon:** Kan inkludere mengden prosessortid og klokkeid brukt, tidsbegrensninger, bokføringsnumre osv.

Prosesskontrollblokk

Informasjon fra denne listen er lagret i en datastruktur, typisk kalt en prosesskontrollblokk (*process control block*), som er opprettet og håndtert av operativ systemet. Det viktigste punktet ved prosesskontrollblokken er at den inneholder tilstrekkelig informasjon til at det er mulig å avbryte en kjørende prosess for så senere fortsette execution som om avbruddet aldri skjedde. Prosesskontrollblokken er nøkkelen som lar OS-et støtte flere prosesser og tilby multiprosesser. Når en prosess blir avbrutt blir verdiene i programtelleren og kontekstdata fra prosessorregistrene lagret i passende felter av den korresponderende prosesskontrollblokken, og tilstanden av prosessen blir endret

til en annen verdi, som *blokkert* eller *klar*. OS-et står da fritt til å sette andre prosesser i kjøretilstand. Programtelleren og kontekstdata for denne prosessen blir da lastet inn i prosessorregistre, og denne prosessen kan da starte å execute.

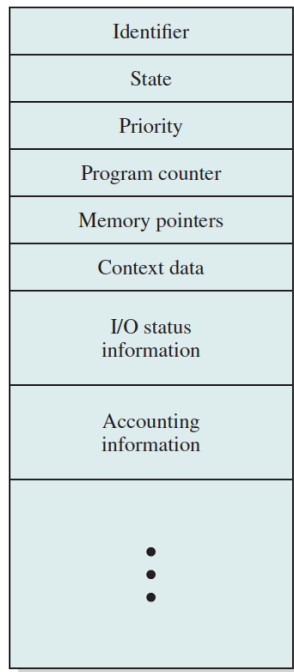


Figure 3.1 Simplified Process Control Block

Kort fortalt:

En prosess består av programkode og assosiert data, pluss en prosesskontrollblokk. Prosesskontrollblokken er den viktigste datastrukturen i et OS. For en enkel-prosessor computer er det ved ethvert tidspunkt på det meste én prosess som executes, og den prosessen er i en *kjørende* tilstand.