

11.4 I/O buffering

Tuesday, April 28, 2020 10:23 PM

For å unngå overhead og å øke effektivitet kan det være nyttig å utføre input-overføringer før en forespørsel gjøres og utføre output-overføringer etter at forespørslene er gjort. Dette kalles *buffering*. Uten I/O-buffering vil det oppstå blokkering av enheter og prosesser ved utførelse og under swapping. Buffere hjelper å jevne ut variasjonen i I/O-bruk mellom prosesser. Full utnyttelse av bufferen får man i systemer hvor det varierer mellom I/O-oppgaver og andre prosess-oppgaver. Bufferen vil da skjule hastighetsvariasjonene til de ulike oppgavene. Et eksempel er ved multiprogrammerings-systemer hvor bufferen øker ytelsen til de individuelle prosessene. Når det kun er I/O-oppgaver vil forsinkelsen etterhvert ta igjen tiden spart i bufferen, og man må uansett vente.

Vi skiller mellom to typer I/O-enheter:

- **Blokk-orienterte enheter:** Holder informasjon i form av blokker som vanligvis har en bestemt størrelse.
- **Strøm-orienterte enheter:** Overfører data inn og ut av enheten som en bytesstrøm, uten blokkstruktur.

Uten buffer

Uten bruk av buffer oppstår det blokkering ved utførelse og bytting av prosesser.

Singel-buffer

Singen-buffer unngår blokkering ved utførelse og bytting. Et enkelt buffer fungerer ved at input-overføringer gjøres til bufferen først. Dette heter *reading ahead* eller *forventet input* fordi det gjøres med forventningen om at det til slutt vil være behov for blokken som avleses. Dette er rimelig, ettersom data ofte aksesseres sekvensielt. Resultatet er økt hastighet, men med en mer komplisert logikk.

Dobbel-buffer

Dobbel-buffer er en forbedring av singel-buffer og utjevner mindre hastighetsvariasjoner. En prosess kan overføre data til og fra et buffer, mens OS-et overfører til og fra et annet buffer. Kostnaden er også her at det er mer sammensatt og kompleks.

Sirkulært-buffer

Sirkulær-bufring utjevner større hastighetsvariasjoner. Dobbel bufring kan være tilstrekkelig dersom prosessen utfører hyppige utbrudd med I/O. Vi kan da benytte mer enn to buffere, hvor hvert buffer er en enhet i den sirkulære bufferen. Dette er samme modell som i *bounded-buffer producer/consumer* fra Kap 5.

Kort fortalt

Buffering er en teknikk som jevner ut toppene i I/O-etterspørsel. Likevel finnes det ikke en mengde buffering som tillater en I/O-enhet å holde følge med en prosess forevig når den gjennomsnittlige etterspørselen av en prosess er høyere en I/O-enheten kan tjenestegjøre. Selv med flere sirkulært-buffere vil alle buffere til slutt fylles opp, og prosessen vil i så fall måtte vente etter prosessering av hver datamengde. Likevel er dette en god løsning for multiprogrammerende-systemer der det er mye variasjon i I/O-aktivitet og flere prosesser å tjenestegjøre – det øker effektiviteten av OS-et og ytelsen på de individuelle prosessorene.

Figuren under demonstrerer hver type buffer:

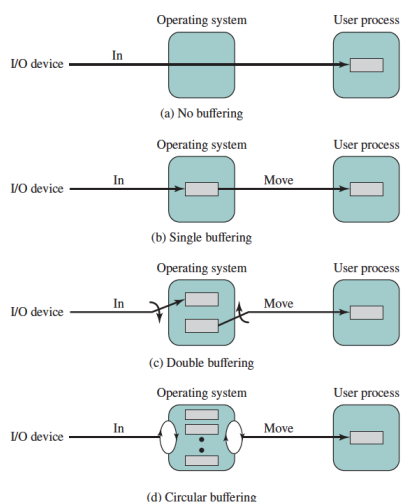


Figure 11.5 I/O Buffering Schemes (Input)

