

12.7 Secondary storage management

Wednesday, April 29, 2020 6:29 PM

I sekundærminnet består en fil av en samling av blokker. Operativsystemet eller filhåndteringssystemet er ansvarlig for tildelingen av blokker til filer. Dette skaper to utfordringer; plass i sekundærminnet må tildeles filene, og det er nødvendig å holde oversikt over plass i sekundærminnet som er tilgjengelig for tildeling av filer. Disse to problemene er relaterte.

Fil-tildeling

Ved fil-tildeling må man ta hensyn til tre ting:

1. **Tidspunkt:** Om tildelingen skal skje i forkant eller ved behov.
2. **Enhetstype:** Om størrelsen på enheten som tildeles en fil skal være bestemt eller variabel.
3. **Datastruktur:** Hvilken datastruktur som skal brukes til å lagre informasjonen om hvilke deler av sekundærminnet som er tildelt filene.

Preallokering vs. dynamisk tildeling

Ved preallokering må man vite hvor mye plass man har behov for i forkant. I noen tilfeller er dette veldig vanskelig, og som ofte ender med at en tildeler for mye plass enn nødvendig – lite effektivt. Derfor er det ofte en fordel med dynamisk tildeling.

Fast eller varierende plass

Man må bestemme om hele filen skal lagres på samme plass, eller kun en blokk på en plass; eller en kombinasjon av disse. Det er en trade-off. Kontinuitet av plass øker ytelse og mange små blokker øker størrelsen på tabellene som skal holde informasjonen. Fast størrelse gjør det enklere å tildele på nytt, mens varierende størrelser utnytter plassen bedre. Det er to hovedmetoder:

1. **Varierende, store kontinuerlige deler:** Gir god ytelse, men er vanskelig å gjenbruke.
 2. **Blokker:** Gir mer fleksibilitet, men tar opp mye plass. Ved varierende lengde kan man benytte ulike disipliner som; *first-fit*, *best-fit* eller *nearest-fit* for plassdeling.
- First-fit: Velger den første ubrukte sammenhengende gruppen av blokker med tilstrekkelig størrelse fra en ledig-blokk-liste.
 - Best-fit: Velger den minste ubrukte gruppen av blokker som er av tilstrekkelig størrelse.
 - Nearest-fit: Velger den ubrukte gruppen av blokker som er av tilstrekkelig størrelse og som er nærmest den foregående filen som ble tildelt (benytter lokalitetsprinsippet).

Det er ikke tydelig hvilken disiplin som er best.

Fil-tildelingsmetoder

Det er hovedsakling tre metoder i vanlig bruk: sammenhengende, kjedet, og indeksert tildeling.

Sammenhengende tildeling

Sammenhengende tildeling er en preallokerings-metode som tildeler et enkelt sett med blokker til en fil ved opprettelse. Denne metoden unngår mye overhead i allokeringstabellen fordi den kun trenger en plass per fil. I tillegg gjør den at aksessering av en fil er enkelt siden alt ligger etter hverandre. Ulempen er ekstern fragmentering, altså at det ikke nødvendigvis er plass til å legge en fil mellom andre filer, og det dermed blir stående mye ledig plass. En annen ulempe er at en må vite størrelsen på forhånd.

Bruken av en sammenpressing-algoritme fra tid til annen vil samle de forskjellige filene og redusere den eksterne fragmenteringen ved å frigjøre plass. Dette er blitt forklart tidligere hvordan fungerer.

Figurene under viser sammenhengende tildeling med og uten sammenpressing.

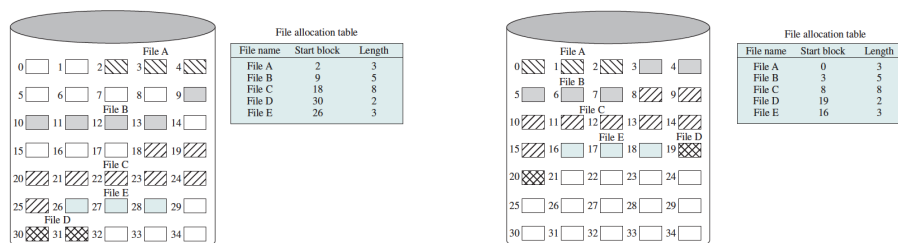


Figure 12.9 Contiguous File Allocation

Figure 12.10 Contiguous File Allocation (After Compaction)

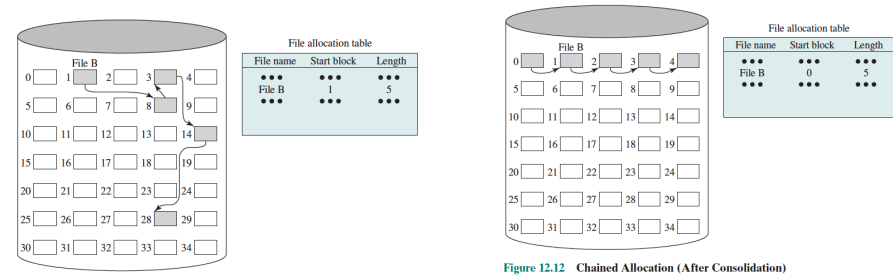
Kjedet tildeling

På den andre siden av skalaen har vi kjedet tildeling. Her gjøres allokering vanligvis på individuelt blokknivå. Kjedet tildeling tillater både preallokering og dynamisk tildeling (som også er mest vanlig). Det gjør det enkelt hvis du skal aksessere en spesifikk blokk, men allokeringstabellen må holde på samme

informasjon om inngangene til hver fil, som viser startblokk og lengde på filen. Seleksjonen av blokker er forholdsvis enkel: enhver ledig blokk kan legges til i kjeden. Ettersom dataen lagres som en lenket liste krever det sekvensielle søk som vi vet er tidkrevende. Dette gjør at kjedet tildeling passer best for sekvensielle filer. En fordel er at det ikke gir ekstern fragmentering (ettersom kun en blokk trengs om gangen), og dermed god utnyttelse av plass.

En konsekvens av kjedet tildeling er at lokalitetsprinsippet ikke blir utnyttet. Man må aksessere mange deler av disken selv om man ønsker etterfølgende blokker. For å overkomme dette kan man innføre en periodisk algoritme som filer – dette gir færre diskaksesser.

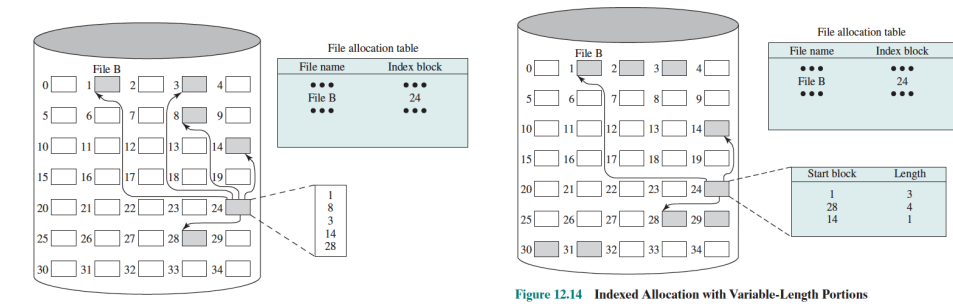
Figurene under viser kjedet tildeling med og uten den periodiske algoritmen.



Indeksert tildeling

Indeksert tildeling løser mange av problemene ved sammenhengende og kjedet tildeling. Allokeringstabellen inneholder en indeks med én kolonne for hver fil; indeksen holder på inngangen til ulike deler tildelt filen. Disse lagres oftest i en egen blokk som allokeringstabellen peker til. Tildelingen kan være basert på både bestemt blokk-lengde og varierende del-størrelser. Tildeling med bestemt blokk-lengde eliminerer ekstern fragmentering, mens varierende lengde utnytter lokalitetsprinsippet. Indeksert tildeling støtter både sekvensiell og direkte aksess til filer, og er derfor den vanligste fil-tildelingsmetoden.

Figurene under viser indeksert tildeling med henholdsvis bestemt blokk-lengde og varierende del-størrelser.



Figuren under viser sammenligningen av de ulike metodene.

Diskallokering - sammenligning

Aspekt	Sammenhengende	Kjedet - fast	Kjedet - variabel	Indeksert - fast	Indeksert - variabel
Tidspunkt	I forkant	I forkant / Ved behov	I forkant / Ved behov	I forkant / Ved behov	I forkant / Ved behov
Enhetsstype	Variabel	Fast	Variabel	Fast	Variabel
Størrelse	Stor	Liten	Middels	Liten	Middels
Frekvens	Minimal	Høy	Middels	Høy	Middels
Plass	Minimal	Liten	Middels	Stor	Middels
Tid	Middels	Lang	Middels	Kort	Middels

Tid for allokering:
Fast: Lange lenker
Variabel: Korte lenker

Plass for allokering:
Fast: Små indeksinnslag
Variabel: Store indeksinnslag

Håndtering av ledig plass

På samme måte som plass tildelt filer må håndteres, så må også ubrukt plass som for øyeblikket ikke blir tildelt noen filer håndteres. For å kunne tildele plass til filer må vi vite hvilken plass som er ledig. Dette gjøres med en disk-allokeringstabell, i tillegg til en fil-allokeringstabell.

Bit-tabeller

Bruker en vektor med en bit for hver blokk på disken. Der 0 representerer tilgjengelig og 1 utilgjengelig.

Denne tabellen tar relativt liten plass og gjør det lett å søke etter sammenhengende og diskrete plasser. Det kan uansett ta en del plass, og sekvensielle søk er tidkrevende.

Kjedede ledige plasser

Ledige plasser kan være kjedet sammen ved en peker og en lengdeverdi for hver ledig plass. Dette krever neglisjerbart med overhead fordi det er ikke behov for en disk-allokeringstabell, men vil kunne gi fragmentering i tillegg til at søk er tidkrevende fordi man må lese hver blokk for å finne pekeren til neste ledige blokk.

Indeksering

Indekseringsmetoden behandler ledige plasser som indekser i en fil og bruker en indeks-tabell lik den i fil-tildeling. For effektivitet burde denne brukes på plasser med varierende lengde. Det gis en plass i tabellen med inngangen for hver ledige del på disken. Indeksering er effektivt.

Ledige blokker-liste

Hver blokk får et tall og listen av ledige blokker opprettholdes i en reservert del av disken.

Volumer

Volumer er en samling av adresserbare sektorer i sekundærminnet som et OS eller en applikasjon kan bruke til datalagring. Sektorene i volumet må ikke være sammenhengende på den fysiske lagringsdisken, men de må fremstå slik for OS-et eller applikasjonen. Et volum kan være resultatet av en demontering og sammenslåing av minde volumer.

Pålitelighet

Det kan oppstå errorer når systemet krasjer. For å sikre pålitelighet kan man gjøre flggende steg når en fil-allokering etterspørres:

1. Lås disk-allokeringstabellen på disken. Dette forhindrer at andre brukere kan endre på tabellen inntil allokeringen er fullført.
2. Søk disk-allokeringstabellen for tilgjengelig plass. Dette antar at det alltid er en kopi av disk-allokeringstabellen i hovedminnet.
3. Tildel plass, og oppdater disk-allokeringstabellen og oppdater disken.
4. Oppdater fil-allokeringstabellen og oppdater disken.
5. Lås opp disk-allokeringstabellen.