Task 3, oblig 1

Victor Florin Moiceanu, Andreas Langeland og Anila Syvertsen

**First-Come First-Serve** er en ikke-preemptiv scheduling-algoritme der prosessene blir behandlet i den rekkefølgen de ankommer køen. Den første prosessen som kommer inn, får først tilgang til CPU-en og får lov til å kjøre til den er ferdig eller inn i et ventetilsand. Denne algoritmen kan føre til høy ventetid for prosesser med lang kjøretid. Det kan også føre til "ventetidsinversjon" der en rask prosess blir blokkert av en tregere prosess som ligger foran i køen. Dette er ikke en god algoritme for time-sharing system, hvor det er viktig at hvert element/bruker får en del av CPU-en ved jevne intervall.

**Round Robin** er en pre-emptiv scheduling-algoritme der prosessene blir tildelt en liten tidsenhet (kvantum) på CPU-en, vanligvis en syklisk rekkefølge. Hvis prosessen ikke er ferdig når kvantumet er utløpt, blir den satt bakerst i køen igjen. RR sikrer rettferdig bruk av CPU-tid og reduserer ventetid for små prosesser, men kan føre til øket administrativ overhead på grunn av hyppig bytter mellom prosesser. RR brukes ofte i interaktive og tidsdelte systemer der ressursene må deles mellom flere brukere eller prosesser. Dette sikrer at alle prosesser får jevn tilgang til CPU-en og er spesielt nyttig når prosessene har lignede prioritet.

Under er tabell med de prosessene som skal utfører med FCFS- og RR-scheduling:

|  |  |  |
| --- | --- | --- |
| **Process** | Arrival time | Burst TIme |
| T1 | 0 | 1 |
| T2 | 0 | 2 |
| T3 | 0 | 4 |
| T4 | 0 | 6 |
| T5 | 0 | 8 |
| T6 | 11 | 8 |
| T7 | 11 | 6 |
| T8 | 11 | 4 |
| T9 | 11 | 2 |
| T10 | 11 | 1 |

Bruker tabellen over og regner ut ventetiden for prosessene med FCFS.scheduling. Bruker formelen: Ventetid = Turnaround Time – Arrival Time – Burst Time

Tabellen under viser resultatet av utregningene

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **Arrival Time** | **Burst Time** | **Finish Time** | **Turnaround Time** | **Waiting Time** |
| T1 | 0 | 1 | 1 | 1 | 0 |
| T2 | 0 | 2 | 3 | 3 | 1 |
| T3 | 0 | 4 | 7 | 7 | 3 |
| T4 | 0 | 6 | 13 | 13 | 7 |
| T5 | 0 | 8 | 21 | 21 | 13 |
| T6 | 11 | 8 | 29 | 18 | 10 |
| T7 | 11 | 6 | 35 | 24 | 18 |
| T8 | 11 | 4 | 39 | 28 | 24 |
| T9 | 11 | 2 | 41 | 30 | 28 |
| T10 | 11 | 1 | 42 | 31 | 30 |

Med verdiene fra tabellen regner vi ut gjennomsnittet for Turnaround Time og Waiting Time

Turnaround Time gjennomsnitt: 176 / 10 = 17.6

Ventetiden gjennomsnitt: 134 / 10 = 13.4

Når vi skal finne hvilken av prosessene som bruker lengst tid er det T10 som har en ventetid på 30 millisekunder. Den prosessen som har kortest tid er det T1 som har en ventetid på 0 millisekunder.

Når vi gjør utregningene for Round Robin scheduling med tids kvantum på 3 brukes formel:

Ventetid for en prosess = (Antall ganger prosessen blir satt tilbake i køen - 1) \* Kvantum

"Antall ganger prosessen blir satt tilbake i køen" er hvor mange ganger prosessen må gå tilbake i køen og vente på sin tur igjen. Dette avhenger av den totale kjøretiden til prosessen og kvantumstørrelsen. "Kvantum" er størrelsen på tidsenheten som hvert prosesskjøringsintervall er delt inn i.

Resultat vist i tabell under:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Prosess** | **Arrival Time** | **Burst Time** | **Finish Time** | **Tournaround Time** | **Waiting Time** |
| T1 | 0 | 1 | 1 | 1 | 0 |
| T2 | 0 | 2 | 3 | 3 | 1 |
| T3 | 0 | 4 | 13 | 13 | 9 |
| T4 | 0 | 6 | 16 | 16 | 10 |
| T5 | 0 | 8 | 40 | 40 | 32 |
| T6 | 11 | 8 | 42 | 31 | 23 |
| T7 | 11 | 6 | 37 | 26 | 20 |
| T8 | 11 | 4 | 38 | 27 | 23 |
| T9 | 11 | 2 | 27 | 16 | 14 |
| T10 | 11 | 1 | 28 | 17 | 16 |

Turnaround Time gjennomsnitt: 190 / 10 = 19

Ventetiden g gjennomsnitt: 148 / 10 = 14.8

Ut fra tabellen ser vi at det er T5 som bruker lengts tid, T5 har en ventetid på 32 enheter. Den prosessen som bruker kortets tid, er T1 som har en ventetid på 0 enheter.

Når vi sammenligner ventetiden med FCFS-scheduling og ventetiden med RR-scheduling ser vi at for disse prosessene er FCFS litt mer effektiv enn RR.