

MODUL PRAKTIKUM 04

TOPIK: SIMULASI ALGORITMA PENJADWALAN CPU & GANTT CHART

Mata Kuliah: Sistem Operasi (INF-201) – Alokasi Waktu: 170 Menit

Program Studi Informatika – Fakultas Teknik – Universitas Bengkulu

1. CAPAIAN PEMBELAJARAN MODUL (SUB-CPMK 4)

Setelah menyelesaikan Modul Praktikum 04 ini, mahasiswa diharapkan mampu:

1. Mengkalkulasi metrik evaluasi penjadwalan CPU: *Turnaround Time* (TAT) dan *Waiting Time* (W).
2. Mensimulasikan eksekusi algoritma penjadwalan FCFS, SJF, dan Round Robin dalam kode program.
3. Membandingkan dampak ukuran *Time Quantum* q terhadap performa Round Robin dan MLFQ.

2. PRASYARAT TUGAS MEMBACA

Sebelum memulai percobaan laboratorium ini, mahasiswa **wajib** telah membaca bab referensi berikut:

- [OSTEP] Chapter 7: *Scheduling: Introduction* (hal. 61–72).
- [OSTEP] Chapter 8: *Scheduling: The Multi-Level Feedback Queue* (hal. 73–84).

3. LANDASAN TEORI RINGKAS

3.1 Metrik Evaluasi Penjadwalan CPU

Dua metrik kuantitatif utama untuk mengevaluasi performa algoritma penjadwalan CPU:

1. **Turnaround Time** (TAT): Total durasi dari waktu submit/kedatangan hingga proses selesai:

$$TAT_i = T_{\text{selesai},i} - T_{\text{datang},i}$$

2. **Waiting Time** (W): Total durasi proses berada di *Ready Queue* menunggu dialokasikan CPU:

$$W_i = TAT_i - \text{Burst Time}_i$$

3.2 5 Aturan Utama Multi-Level Feedback Queue (MLFQ)

1. **Rule 1:** Jika $\text{Priority}(A) > \text{Priority}(B)$, A berjalan (B tidak).
2. **Rule 2:** Jika $\text{Priority}(A) = \text{Priority}(B)$, A & B berjalan dalam Round Robin.
3. **Rule 3:** Saat proses baru memasuki sistem, diletakkan di antrean prioritas tertinggi.
4. **Rule 4:** Setelah proses menghabiskan jatah waktunya di satu tingkat, prioritasnya diturunkan 1 tingkat.
5. **Rule 5 (Priority Boost):** Setelah periode S , pindahkan **semua** proses ke antrean teratas.

4. KODE ACUAN UTUH (EXECUTABLE C LISTING: fcfs_sim.c)

```

1  #include <stdio.h>
2
3  struct Process {
4      int pid;
5      int burst_time;
6      int waiting_time;
7      int turnaround_time;
8  };
9
10 void calculate_times(struct Process proc[], int n) {
11     proc[0].waiting_time = 0;
12     proc[0].turnaround_time = proc[0].burst_time;
13
14     for (int i = 1; i < n; i++) {
15         proc[i].waiting_time = proc[i-1].waiting_time + proc[i-1].burst_time
16         ;
17         proc[i].turnaround_time = proc[i].waiting_time + proc[i].burst_time;
18     }
19 }
20
21 int main() {
22     struct Process proc[] = {{1, 24}, {2, 3}, {3, 3}};
23     int n = sizeof(proc) / sizeof(proc[0]);
24
25     calculate_times(proc, n);
26
27     printf("PID\tBurst\tWaiting\tTurnaround\n");
28     float total_w = 0, total_tat = 0;
29     for (int i = 0; i < n; i++) {
30         total_w += proc[i].waiting_time;
31         total_tat += proc[i].turnaround_time;
32         printf("P%d\t%d\t%d\t%d\n", proc[i].pid, proc[i].burst_time,
33             proc[i].waiting_time, proc[i].turnaround_time);
34     }
35
36     printf("\nRata-rata Waiting Time: %.2f ms\n", total_w / n);
37     printf("Rata-rata Turnaround Time: %.2f ms\n", total_tat / n);
38     return 0;
39 }

```

Listing 1: Program Simulasi Penjadwalan FCFS C (fcfs_sim.c)

5. PROSEDUR LANGKAH KERJA PERCOBAAN

1. Buat berkas fcfs_sim.c dan kompilasi menggunakan gcc fcfs_sim.c -o fcfs_sim.
2. Jalankan program dan catat hasil rata-rata *Waiting Time* untuk urutan kedatangan P_1, P_2, P_3 .
3. Ubah urutan proses menjadi P_2, P_3, P_1 dan amati perubahan rata-rata *Waiting Time* (efek *Convoy Effect*).

6. PANDUAN TROUBLESHOOTING ERROR UMUM

7. TUGAS PRAKTIKUM & PERTANYAAN ANALISIS LAPORAN

Pesan Error / Kendala	Penyebab Utama	Solusi Solutif
Rata-rata waktu bertipe integer (hasil bulat).	Pembagian variabel <code>total_w / n</code> dilakukan dalam integer.	Casting variabel menjadi float: <code>(float)total_w / n</code> .

1. **Pengembangan Kode SJF:** Modifikasi kode C di atas untuk mengimplementasikan algoritma **Shortest Job First (SJF)** non-preemptive!
2. **Analisis MLFQ:** Mengapa Aturan ke-5 (*Priority Boost*) pada MLFQ sangat vital pada sistem komputer nyata?