

Sistem Operasi (INF-201)

Pertemuan 4: Penjadwalan CPU (CPU Scheduling)

Tim Dosen Sistem Operasi

Program Studi Informatika – Universitas Bengkulu

Semester Ganjil 2026/2027



Capaian Pembelajaran (Sub-CPMK 4)

- Mahasiswa mampu menganalisis kriteria performa Penjadwalan CPU.
- Mahasiswa mampu menghitung dan mengevaluasi *Waiting Time* dan *Turnaround Time* pada algoritma FCFS, SJF, Priority, dan Round Robin.
- Mahasiswa memahami 5 Aturan Utama **Multilevel Feedback Queue (MLFQ)** (OSTEP Ch. 8).

Kriteria Evaluasi Penjadwalan CPU (OSTEP Ch. 7)

- **Utilisasi CPU:** Persentase sibuk CPU (max 100%).
- **Throughput:** Jumlah proses yang selesai per satuan waktu.
- **Turnaround Time (TAT):** Selisih waktu selesai dengan waktu datang:

$$TAT = T_{\text{selesai}} - T_{\text{datang}}$$

- **Waiting Time (W):** Total waktu tunggu proses di *Ready Queue*:

$$W = TAT - \text{Burst Time}$$

- **Response Time:** Waktu dari submit hingga respon pertama kali muncul.

Non-Preemptive:

- **FCFS (First-Come, First-Served):**
Sederhana namun rentan terhadap *Convoy Effect*.
- **SJF (Shortest Job First):** Optimal untuk meminimalkan rata-rata *Waiting Time*.

Preemptive:

- **SRTF (Shortest Remaining Time First):**
Versi preemptive dari SJF.
- **Round Robin (RR):** Jatah waktu (*Time Quantum q*).
- **MLFQ:** Multi-queue dinamis berdasarkan riwayat eksekusi.

5 Aturan Utama MLFQ (OSTEP Ch. 8)

Aturan Dasar Multi-Level Feedback Queue

- 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 (terlepas berapa kali melepaskan CPU), prioritasnya diturunkan 1 tingkat.
- ⑤ **Rule 5 (Priority Boost):** Setelah periode waktu S , pindahkan **semua** proses ke antrean paling atas (mencegah *Starvation*).

Tugas Membaca Mingguan (Reading Assignment)

Persiapan Pertemuan 5: Sinkronisasi Proses & Multi-threading

Sebelum mengikuti kuliah minggu depan, mahasiswa wajib membaca:

- **[OSTEP]** Chapter 26: *Concurrency: An Introduction* (hal. 291–304)
- **[OSTEP]** Chapter 28: *Locks* (hal. 319–336)
- **[OSTEP]** Chapter 31: *Semaphores* (hal. 363–378)
- **[Think OS]** Chapter 7–9: *Threads, Mutex, Semaphores* (hal. 55–82)
- **[xv6 RISC-V]** Chapter 6: *Locking* (hal. 63–72)

Pertanyaan Pemicu Diskusi Mandiri

Mengapa instruksi biasa seperti `counter++` di C dapat menimbulkan *Data Race* ketika dijalankan oleh dua thread berbeda?