

Sistem Operasi (INF-201)

Pertemuan 9: Virtual Memory & Page Replacement

Tim Dosen Sistem Operasi

Program Studi Informatika – Universitas Bengkulu

Semester Ganjil 2026/2027



Capaian Pembelajaran (Sub-CPMK 9)

- Mahasiswa mampu menjelaskan mekanisme **Demand Paging**.
- Mahasiswa mampu menghitung **Page Fault Rate**.
- Mahasiswa mampu menganalisis & membandingkan algoritma page replacement (FIFO, LRU, CLOCK/Second Chance).
- Mahasiswa mampu merancang simulasi page replacement.

Virtual Memory

Pemisahan memori logis pengguna dari memori fisik. Mengizinkan eksekusi proses yang ukurannya lebih besar daripada memori fisik (RAM) yang tersedia.

- **Demand Paging:** Membawa sebuah page ke memori hanya saat dibutuhkan (saat ada referensi ke page tersebut).
- Jika page tidak ada di RAM, terjadi **Page Fault**. OS harus mengambilnya dari disk (Swap space).

Penanganan Page Fault (OSTEP Ch. 21)

- 1 Periksa page table internal OS: jika referensi invalid $\implies$ abort, jika tidak ada di RAM $\implies$ page fault.
- 2 Cari frame fisik yang kosong. Jika tidak ada, gunakan **Page Replacement Algorithm** untuk mengosongkan frame.
- 3 Jadwalkan operasi disk untuk membaca page yang diinginkan ke frame kosong tersebut.
- 4 Perbarui page table (set bit valid).
- 5 Mulai ulang instruksi (*Restart instruction*) yang menyebabkan page fault.

Algoritma Page Replacement: FIFO & LRU

- **FIFO (First-In, First-Out):** Mengganti page yang pertama kali dibawa ke dalam memori. Sering mengalami *Belady's Anomaly*.
- **LRU (Least Recently Used):** Mengganti page yang belum digunakan untuk periode waktu terlama. Optimal tapi mahal diimplementasikan secara perangkat keras (butuh *timestamp* atau *stack*).

Contoh Reference String: 7, 0, 1, 2, 0, 3 (Kapasitas = 3 Frames)

- **FIFO:** 7 → 0 → 1 → 2 (menggantikan 7) → 0 (hit) → 3 (menggantikan 0). Total Page Fault = 5.
- **LRU:** 7 → 0 → 1 → 2 (menggantikan 7) → 0 (hit) → 3 (menggantikan 1). Total Page Fault = 4.

Clock Algorithm

Pendekatan LRU (LRU approximation) menggunakan bit referensi (*use bit*).

- Setiap page memiliki *use bit*. Diset ke 1 oleh hardware saat diakses.
- OS menggunakan penunjuk melingkar (*clock hand*). Jika *use bit* = 1, ubah ke 0 (beri kesempatan kedua) dan pindah ke page berikutnya. Jika *use bit* = 0, ganti page ini.

Diskusi: Mengapa OS nyata lebih memilih algoritma Clock dibanding LRU murni?

Thrashing & Working Set Model

- **Thrashing:** Kondisi di mana proses menghabiskan lebih banyak waktu untuk melakukan paging (tukar guling page ke disk) daripada mengeksekusi instruksi.
- Terjadi jika proses tidak memiliki cukup frame.
- **Working Set Model:** Berbasis konsep *locality*. Memastikan OS mengalokasikan cukup frame sebesar jumlah page yang aktif digunakan bersamaan (*working set*).

Tugas Praktikum 3

Mendesain dan memprogram **Page Replacement Simulator** menggunakan bahasa C. (Bobot 6%).

Tugas Membaca Mingguan

- **[OSTEP]** Chapter 21, 22, dan 23.
- **[Think OS]** Chapter 6.