Sistem Operasi (INF-201)

Pertemuan 13: Subsistem I/O & Manajemen Perangkat

Tim Dosen Sistem Operasi

Program Studi Informatika – Universitas Bengkulu

Semester Ganjil 2026/2027

Capaian Pembelajaran (Sub-CPMK 13)

- Mahasiswa mampu menjelaskan mekanisme DMA (Direct Memory Access) dan mengapa diperlukan untuk I/O berkecepatan tinggi.
- Mahasiswa mampu menganalisis teknik buffering, caching, dan spooling pada Kernel I/O subsystem.
- Mahasiswa mampu membandingkan metode Programmed I/O, Interrupt-driven I/O, dan DMA.
- Mahasiswa memahami perbedaan antara perangkat blok (*block devices*) dan karakter (*character devices*).

Perangkat Keras I/O & Klasifikasi

Kategori Perangkat I/O

- **Block Devices:** Mentransfer data dalam blok ukuran tetap (misal: HDD, SSD). Mendukung akses acak (*random access*).
- **Character Devices:** Mentransfer data sebagai aliran byte (*stream of bytes*) tanpa struktur blok (misal: Keyboard, Mouse, Serial Port).
- **Network Devices:** Menangani transfer paket jaringan (Socket API).

Komunikasi CPU dengan perangkat keras menggunakan *Ports*, *Buses* (PCIe, USB), dan *Device Controllers*.

Metode Transfer I/O

Programmed I/O (PIO)

- CPU secara terus-menerus mengecek status perangkat (*polling* / *busy-waiting*).
- Sangat membebani CPU, utilisasi CPU 100% selama transfer.

Interrupt-driven I/O

- CPU memberikan perintah I/O lalu mengerjakan hal lain. Perangkat memicu *Interrupt* saat selesai.
- Masalah: Terlalu banyak interrupt untuk transfer data besar (1 interrupt per byte/word).

Direct Memory Access (DMA)

- DMA Controller menangani transfer blok data langsung antara Perangkat I/O dan RAM tanpa intervensi CPU.
- CPU hanya menerima 1 interrupt setelah seluruh blok selesai ditransfer.

Mekanisme DMA (Direct Memory Access)

Langkah-Langkah DMA Transfer

1. CPU memprogram DMA Controller dengan memberikan alamat sumber, alamat tujuan, dan jumlah byte (misal, untuk membaca disk ke RAM).
2. CPU melanjutkan eksekusi proses lain.
3. DMA mengambil alih *memory bus* dan mentransfer data langsung dari Disk Controller ke RAM.
4. Setelah selesai, DMA mengirimkan sinyal Interrupt ke CPU.

Pseudo-code CPU saat setup DMA:

```
dma_controller->source = DISK_BUFFER;
dma_controller->destination = RAM_ADDRESS;
dma_controller->bytes = 4096;
dma_controller->command = START_TRANSFER;
// CPU langsung bisa melakukan context switch ke thread lain
```

Kernel I/O Subsystem Services

- **Buffering:** Menyimpan sementara data yang sedang ditransfer untuk mengatasi perbedaan kecepatan antara perangkat dan aplikasi, atau perbedaan ukuran transfer.
- **Caching:** Menyimpan salinan data I/O di memori utama yang lebih cepat untuk akses masa depan (*Page Cache*).
- **Spooling (Simultaneous Peripheral Operations On-Line):** Antrean output untuk perangkat yang tidak bisa membagi layanannya (misal: Printer).
- **Device Reservation:** Menjamin akses eksklusif ke perangkat tertentu agar proses tidak saling mengganggu (mencegah *interleaving*).

Tugas Membaca Mingguan (Reading Assignment)

Persiapan Pertemuan 14: Keamanan & Proteksi Kernel

Sebelum mengikuti kuliah minggu depan, mahasiswa wajib membaca:

- **[OSTEP]** Chapter 54: *Security: Introduction & Protection Mechanisms*
- **[TLCL]** Chapter 9: *Permissions*

Pertanyaan Pemicu Diskusi Mandiri

Bagaimana fitur *pledge* dan *unveil* pada OpenBSD membatasi akses system call dan membatasi direktori yang dapat diakses oleh proses aplikasi?