

PANDUAN PRAKTIKUM SISTEM OPERASI

PARUH PERTAMA PERKULIAHAN (MINGGU 1 S.D. 7)

KODE MATA KULIAH: INF-201 (3 SKS: 2 TEORI / 1 PRAKTIKUM)

PROGRAM STUDI INFORMATIKA – FAKULTAS TEKNIK – UNIVERSITAS BENGKULU

1. KETENTUAN UMUM & LINGKUNGAN PRAKTIKUM

1. **Lingkungan Sistem:** Praktikum dilaksanakan sepenuhnya di lingkungan sistem operasi **Linux** (**Ubuntu/Debian/Fedora**) dengan antarmuka *Command Line Interface* (CLI).
2. **Perangkat Lunak Utama:** Kompiler `gcc/clang`, utilitas `make`, debugger `gdb`, pelacak system call `strace`, dan analisis thread `ThreadSanitizer`.
3. **Bobot Evaluasi:** Nilai Praktikum berkontribusi sebesar **17%** dari Nilai Akhir Mata Kuliah yang terbagi menjadi:
 - Tugas Terstruktur 1 (Minggu 3 – Custom Linux Shell C Program): **5%**
 - Tugas Terstruktur 2 (Minggu 5 – POSIX Thread Synchronization): **6%**
 - Laporan & Aktifitas Mingguan: **6%**

2. RINCIAN MODUL PRAKTIKUM MINGGU 1 S.D. 7

2.1 Modul 1: Pengenalan Linux CLI & Pelacakan System Calls (`strace`)

Acuan Buku: *TLCL Ch. 1–2; OSTEP Ch. 6 (Limited Direct Execution)*.

Tujuan: Mahasiswa mampu bernavigasi di terminal Linux dan menganalisis panggilan sistem (*System Calls*) yang dipanggil oleh aplikasi ke kernel.

1. Jalankan utilitas `strace` untuk menganalisis eksekusi perintah `ls`:

```
1 $ strace -c ls
2 $ strace -e trace=openat,read,write,close cat /etc/hosts
```

2. Catat jumlah dan durasi *system call* utama seperti `read`, `write`, `openat`, dan `mmap`.

2.2 Modul 2: Inspeksi Kernel & Scripting Bash Dasar

Acuan Buku: *TLCL Ch. 24; xv6 RISC-V Ch. 2*.

Tujuan: Mahasiswa memahami struktur direktori maya `/proc` dan alur booting Linux.

1. Inspeksi informasi kernel dan proses pertama (PID 1):

```
1 $ uname -a
2 $ cat /proc/cpuinfo | grep "model_name"
3 $ ps -p 1 -o comm=
```

2. Buatlah sebuah script bash `sysinfo.sh` yang menampilkan memori bebas dan penggunaan disk secara otomatis.

2.3 Modul 3: Tugas Terstruktur 1 – Custom Linux Shell (myshell.c)

Acuan Buku: *OSTEP Ch. 5 (Process API); xv6 RISC-V Ch. 1.*

Tujuan: Merancang terminal shell interaktif dalam bahasa C menggunakan POSIX System Calls.

1. Program wajib mengimplementasikan `fork()`, `execvp()`, dan `waitpid()`.
2. Potongan kode alur utama eksekusi proses anak:

```
1 pid_t pid = fork();
2 if (pid == 0) {
3     // Child Process: Jalankan perintah CLI
4     execvp(args[0], args);
5     perror("execvp failed");
6     exit(1);
7 } else if (pid > 0) {
8     // Parent Process: Tunggu anak selesai
9     waitpid(pid, NULL, 0);
10 }
```

3. Tangani perintah built-in `cd` (`chdir()`) dan `exit`.

2.4 Modul 4: Simulasi Algoritma Penjadwalan CPU

Acuan Buku: *OSTEP Ch. 7 & 8 (CPU Scheduling & MLFQ).*

Tujuan: Mensimulasikan dan membandingkan performa algoritma FCFS, SJF, Round Robin, dan MLFQ.

1. Hitung *Turnaround Time* ($TAT = T_{\text{selesai}} - T_{\text{datang}}$) dan *Waiting Time* ($W = TAT - T_{\text{burst}}$).
2. Buatlah simulasi visual Gantt Chart eksekusi proses untuk mengamati *Convoy Effect* pada FCFS vs efisiensi SJF.

2.5 Modul 5: Tugas Terstruktur 2 – Thread Synchronization & ThreadSanitizer

Acuan Buku: *OSTEP Ch. 26, 28, 31; Think OS Ch. 7–9.*

Tujuan: Memecahkan masalah *Race Condition* menggunakan POSIX Threads, Mutex, dan Semaphores.

1. Implementasikan masalah *Producer-Consumer* menggunakan `pthread_mutex_t` dan `sem_t`.
2. Kompilasi dan verifikasi kebebasan *data race* menggunakan ThreadSanitizer:

```
1 $ gcc -fsanitize=thread -g -pthread producer_consumer.c -o demo
2 $ ./demo
```

2.6 Modul 6: Simulasi Banker's Algorithm (Deadlock Avoidance)

Acuan Buku: *OSTEP Ch. 32 (Common Concurrency Bugs).*

Tujuan: Menerapkan logika Banker's Algorithm untuk menentukan *Safe State*.

1. Hitung matriks kebutuhan $Need_i = Max_i - Allocation_i$.
2. Tentukan urutan eksekusi aman $\langle P_0, P_1, \dots, P_n \rangle$ berdasarkan ketersediaan vektor Work.

2.7 Modul 7: Simulasi Paging Memori & Hitung EAT TLB

Acuan Buku: *OSTEP Ch. 18–20; xv6 RISC-V Ch. 3.*

Tujuan: Mensimulasikan translasi alamat Paging $(p, d) \rightarrow (f, d)$ dan memformulasi Effective Access Time (EAT).

1. Hitung EAT berdasarkan variasi TLB Hit Ratio α :

$$\text{EAT} = \alpha(\epsilon + t_m) + (1 - \alpha)(\epsilon + 2t_m)$$

2. Analisis efisiensi 4-Level Page Table pada arsitektur x86-64.

3. RUBRIK PENILAIAN PRAKTIKUM

Kriteria Penilaian	Bobot	Deskripsi Indikator Keberhasilan
Kebenaran Logika & System Calls	40%	Program berhasil dieksekusi tanpa error logis/crash.
Manajemen Memori (No Leak)	20%	Pengalokasian memori terbebas dari <i>memory leak</i> .
Sinkronisasi (ThreadSanitizer Clean)	20%	Eksekusi thread terbukti bebas dari <i>Data Race</i> .
Kerapian Kode & Laporan Mingguan	20%	Mengikuti standar koding C/Linux kernel dan laporan rapi.

4. LEMBAR PENGESAHAN & TEMPLATE LAPORAN PRAKTIKUM

Menyetujui,
Koordinator Praktikum SO

Bengkulu, 12 Agustus 2026
Dosen Pengampu Mata Kuliah

Tim Laboratorium Informatika
NIP. 19850101 201012 1 001

Tim Dosen Sistem Operasi
NIP. 19880202 201503 1 002

Ketentuan Format Laporan Praktikum Mahasiswa:

Setiap laporan praktikum diketik dengan format PDF memuat: (1) Halaman Judul & Identitas; (2) Ringkasan Teori; (3) Tangkapan Layar Eksekusi Kode & Terminal; (4) Kode Sumber C/Bash Lengkap; (5) Pembahasan & Analisis Jawaban Pertanyaan Modul.