

Operating System Course Report - First Half of the Semester

B class

October 1, 2024

Contents

1	Introduction	3
2	Course Overview	3
2.1	Objectives	3
2.2	Course Structure	3
3	Topics Covered	4
3.1	Basic Concepts and Components of Computer Systems	4
3.2	System Performance and Metrics	4
3.3	System Architecture of Computer Systems	4
3.4	Process Description and Control	4
3.5	Scheduling Algorithms	5
3.6	Process Creation and Termination	5
3.7	Introduction to Threads	5
3.8	File System	5
3.8.1	File System Structure	7

1 Introduction

This report summarizes the topics covered during the first half of the Operating System course. It includes theoretical concepts, practical implementations, and assignments. The course focuses on the fundamentals of operating systems, including system architecture, process management, CPU scheduling, and deadlock handling.

2 Course Overview

2.1 Objectives

The main objectives of this course are:

- To understand the basic components and architecture of a computer system.
- To learn process management, scheduling, and inter-process communication.
- To explore file systems, input/output management, and virtualization.
- To study the prevention and handling of deadlocks in operating systems.

2.2 Course Structure

The course is divided into two halves. This report focuses on the first half, which covers:

- Basic Concepts and Components of Computer Systems
- System Performance and Metrics
- System Architecture of Computer Systems
- Process Description and Control
- Scheduling Algorithms
- Process Creation and Termination

- Introduction to Threads
- File Systems
- Input and Output Management
- Deadlock Introduction and Prevention
- User Interface Management
- Virtualization in Operating Systems

3 Topics Covered

3.1 Basic Concepts and Components of Computer Systems

This section explains the fundamental components that make up a computer system, including the CPU, memory, storage, and input/output devices.

3.2 System Performance and Metrics

This section introduces various system performance metrics used to measure the efficiency of a computer system, including throughput, response time, and utilization.

3.3 System Architecture of Computer Systems

Describes the architecture of modern computer systems, focusing on the interaction between hardware and the operating system.

3.4 Process Description and Control

Processes are a central concept in operating systems. This section covers:

- Process states and state transitions
- Process control block (PCB)
- Context switching

3.5 Scheduling Algorithms

This section covers:

- First-Come, First-Served (FCFS)
- Shortest Job Next (SJN)
- Round Robin (RR)

It explains how these algorithms are used to allocate CPU time to processes.

3.6 Process Creation and Termination

Details how processes are created and terminated by the operating system, including:

- Process spawning
- Process termination conditions

3.7 Introduction to Threads

This section introduces the concept of threads and their relation to processes, covering:

- Single-threaded vs. multi-threaded processes
- Benefits of multithreading

3.8 File System

File adalah unit penyimpanan logika yang diabstraksi oleh sistem operasi dari perangkat penyimpanan. File berisi informasi yang disimpan pada penyimpanan sekunder (seperti *magnetic disk*, *magnetic tape*, dan *optical disk*). Informasi dalam file didefinisikan oleh pembuatnya. Setiap file memiliki struktur tertentu tergantung pada isinya. File dapat terdiri dari data (baik berupa data numerik, karakter, maupun biner) serta program seperti *source program*, *object program*, dan *executable program*.

a. Ruang Alamat Logis yang Berdekatan

File dalam sistem biasanya disimpan sebagai ruang alamat logis yang berdekatan. Hal ini berarti bahwa file menempati urutan alamat logis dalam memori. Namun, hal ini tidak selalu menyiratkan bahwa file juga menempati ruang fisik yang berdekatan pada disk. Sistem file berfungsi untuk menyederhanakan akses dan pengambilan file dengan membuatnya tampak berurutan, terlepas dari bagaimana penyimpanannya secara fisik.

b. Jenis File

File dapat dikategorikan menjadi dua jenis utama:

1. **File Data:** Berisi data terstruktur atau tidak terstruktur dan dapat dikategorikan lebih lanjut menjadi:
 - **Numerik:** Data yang direpresentasikan sebagai angka (misalnya, data statistik atau data keuangan).
 - **Character:** Data yang direpresentasikan sebagai teks yang dapat dibaca manusia.
 - **Biner:** File non-teks, seperti file media (audio, video), atau jenis data lainnya yang dikodekan.
2. **File Program:** File yang berisi kode yang dapat dieksekusi untuk menjalankan aplikasi atau layanan pada komputer.

c. Konten File yang Ditentukan oleh Pencipta

Isi file ditentukan oleh pembuatnya dan tujuan dari pembuatan file tersebut. File yang berbeda mungkin berisi instruksi (*kode sumber*) untuk program komputer, dokumentasi tertulis (file teks), atau kode yang sudah dikompilasi (file *executable*). Contoh jenis file:

- **File Teks:** File sederhana yang berisi karakter yang dapat dibaca oleh manusia.
- **File Sumber:** File yang berisi kode sumber yang ditulis dalam bahasa pemrograman.
- **File yang dapat dieksekusi:** File yang dapat dijalankan langsung oleh sistem operasi sebagai program. File ini berisi kode yang sudah dikompilasi atau ditafsirkan.

3.8.1 File System Structure

Struktur sistem file mengatur bagaimana file disimpan, diakses, dan diatur dalam perangkat penyimpanan. Struktur ini mencakup beberapa komponen, seperti direktori, tabel alokasi, dan *metadata* yang membantu mengelola file dengan efisien. Struktur file dapat dikategorikan ke dalam berbagai jenis berdasarkan kompleksitas organisasi data. Misalnya:

1. **Simple record Structure:** terdiri dari urutan baris atau catatan dengan panjang tetap atau variabel
2. **Complex structures:** lebih rumit mencakup dokumen yang diformat dan file beban yang dapat direlokasi, yang memerlukan mekanisme penanganan yang lebih canggih.

Struktur ini sangat penting dalam mengelola bagaimana data disimpan dan diakses, dan karakter kontrol dapat mensimulasikan struktur yang kompleks. Keputusan tentang organisasi struktur file dipengaruhi oleh sistem operasi dan program yang digunakan. Tipe file juga digunakan untuk menunjukkan struktur internal dari file. File tertentu harus konfirmasi ke struktur yang dibutuhkan yang dimengerti oleh sistem operasi. Misalnya, sistem operasi membutuhkan file *executable* yang mempunyai struktur khusus sehingga dapat menentukan dimana letak memori dan lokasi dari instruksi pertama.

Beberapa sistem operasi menggunakan sekumpulan sistem pendukung struktur file dengan sejumlah operasi khusus untuk manipulasi file dengan struktur tersebut. Hal ini menjadi kelemahan pada sistem operasi yang mendukung struktur file lebih dari satu. Jika sistem operasi menentukan 10 struktur file berbeda, maka perlu menyertakan kode untuk mendukung struktur file tersebut. Setiap file perlu dapat didefinisikan sebagai satu dari tipe file yang didukung oleh sistem operasi. Beberapa sistem operasi seperti UNIX dan MS-DOS hanya mendukung sejumlah struktur file. UNIX menentukan setiap file merupakan deret 8 bit byte dan bit tersebut tidak diterjemahkan oleh sistem operasi. Skema ini mempunyai fleksibilitas maksimum, tetapi sedikit dukungan. Setiap program aplikasi harus menyertakan kode sendiri untuk menterjemahkan file input ke dalam struktur yang tepat. Setidaknya semua SO harus mendukung sedikitnya satu struktur file *executable* sehingga sistem dapat *load* dan menjalankan program.

a. File Attributes

File memiliki nama dan data. Selain itu, ia juga menyimpan informasi meta seperti tanggal dan waktu pembuatan file, ukuran saat ini, tanggal terakhir dimodifikasi, dll. Semua informasi ini disebut atribut sistem file. Berikut adalah beberapa atribut File penting yang digunakan di OS:.

1. **Nama:** Ini adalah satu-satunya informasi yang disimpan dalam bentuk yang dapat dibaca manusia.
2. **Pengidentifikasi:** Setiap file diidentifikasi dengan nomor tag unik dalam sistem file yang dikenal sebagai pengidentifikasi.
3. **Tempat:** Menunjuk ke lokasi file di perangkat.
4. **Jenis:** Atribut ini diperlukan untuk sistem yang mendukung berbagai jenis file.
5. **Ukuran:** Atribut yang digunakan untuk menampilkan ukuran file saat ini.
6. **Perlindungan:** Atribut ini menetapkan dan mengontrol hak akses membaca, menulis, dan mengeksekusi file.
7. **Waktu, tanggal, dan keamanan:** Ini digunakan untuk perlindungan, keamanan, dan juga digunakan untuk pemantauan

Atribut ini disimpan dalam struktur direktori sistem file, yang memungkinkan sistem operasi untuk mengelola file dengan lebih efisien.

b. File Operations

1. File sebagai Tipe Data Abstrak:

File adalah konsep dalam komputasi yang mewakili kumpulan data yang disimpan secara terstruktur. Ini diperlakukan sebagai tipe data abstrak (ADT) karena cara data disimpan dan diakses disembunyikan dari pengguna. Sistem mengabstraksi kompleksitas mengakses dan mengelola data, menyediakan struktur logis bagi pengguna untuk berinteraksi dengan file.

2. **Buat (Create):**

Pembuatan file dalam sistem file melibatkan alokasi ruang dan menginisialisasi metadata file. Metadata mencakup nama file, jenis, ukuran, dan atribut lain yang diperlukan agar file dikelola dalam sistem. File yang baru dibuat mungkin kosong pada awalnya dan siap untuk ditulis.

3. **Tulis (Write):**

Menulis ke file berarti memasukkan data ke dalam file, dimulai dari lokasi penunjuk tulis saat ini. Proses ini menambahkan data baru atau mengganti data yang ada pada posisi yang ditentukan dalam file. Jika file tidak ada, beberapa sistem akan membuat file baru untuk menyimpan data.

4. **Baca (Read):**

Operasi baca mengambil data dari file mulai dari lokasi penunjuk baca saat ini. Penunjuk menunjukkan di mana dalam file sistem harus mulai membaca. Setelah data dibaca, data dimuat ke dalam memori untuk digunakan atau diproses lebih lanjut oleh program.

5. **Reposisi dalam File – Cari:**

Operasi pencarian mengubah posisi penunjuk baca atau tulis dalam file tanpa membaca atau menulis data apa pun. Ini memungkinkan sistem untuk menavigasi ke berbagai bagian file dengan cepat. Misalnya, pindah ke akhir file untuk menambahkan data atau pergi ke lokasi tertentu untuk mengubah informasi yang ada.

6. **Hapus (Delete):**

Operasi hapus menghapus file dari sistem file. Itu dapat menghapus seluruh file atau menandai ruang yang ditempatinya sebagai tersedia untuk data baru, tergantung pada sistem file yang digunakan. Dalam kebanyakan kasus, metadata dan referensi ke file juga dihapus.

7. **Truncate:**

Pemotongan mengurangi ukuran file dengan memotong kontennya dari titik tertentu dan seterusnya. Operasi ini berguna ketika ukuran file perlu dikurangi, sambil tetap menjaga file itu sendiri dan metadatanya tetap utuh.

8. **Open (Buka):**

Saat file dibuka, sistem mengambil metadatanya dari struktur direk-

tori dan memuatnya ke dalam memori. Deskriptor file dibuat, memungkinkan operasi berikutnya (seperti membaca atau menulis) dilakukan. Membuka file juga mengatur penunjuk baca/tulis ke posisi yang sesuai.

9. Close (Tutup):

Menutup file berarti menyelesaikan operasi yang tersisa (seperti mem-bilas data yang di-buffer ke disk) dan menghapus informasi file dari memori. Operasi ini memastikan bahwa tidak ada operasi lebih lanjut yang dapat dilakukan pada file hingga dibuka kembali.

c. Open Files

Manajemen *open files* adalah salah satu elemen penting dalam sistem operasi modern, karena pada dasarnya, setiap proses yang berjalan sering kali berinteraksi dengan file untuk membaca data, menulis informasi, atau mengakses sumber daya tertentu. Oleh karena itu, manajemen *open files* yang efektif akan berdampak langsung pada kinerja beberapa komponen utama yang dibutuhkan untuk mengelola file:

1. Tabel file terbuka:

Struktur yang melacak semua file yang sedang dibuka oleh proses. Setiap entri biasanya berisi informasi tentang file seperti penunjuk file, deskriptor file, dan detail relevan lainnya. Tabel file terbuka sangat penting untuk manajemen file yang efisien dalam sistem operasi. Tanpa tabel ini, OS akan kesulitan melacak file mana yang terbuka, dan oleh proses mana. Tabel ini menyediakan kontrol terpusat untuk penanganan file.

2. Penunjuk berkas:

Ini merujuk pada penunjuk yang menyimpan posisi baca/tulis terakhir dalam berkas. Ini biasanya disimpan per proses untuk melacak posisi tertentu tempat operasi akan dilanjutkan. Penunjuk berkas memungkinkan setiap proses mempertahankan posisinya di dalam berkas secara independen. Hal ini penting untuk lingkungan multitugas di mana beberapa proses dapat membaca atau menulis ke berkas yang sama.

3. Jumlah file yang dibuka:

Penghitung yang digunakan untuk melacak berapa banyak proses yang telah membuka file tertentu. Ini berguna untuk memastikan bahwa

entri file dalam tabel file yang dibuka hanya dihapus setelah proses terakhir menutup file. Penghitungan file yang dibuka membantu memastikan bahwa file tidak dihapus sebelum waktunya dari tabel file yang dibuka saat masih digunakan oleh proses lain. Penghitungan ini menyediakan mekanisme penghitungan referensi yang sederhana namun efektif.

4. Lokasi disk berkas:

Cache yang menyimpan informasi mengenai lokasi fisik berkas pada disk, yang memungkinkan akses lebih cepat selama operasi. Caching lokasi disk mengurangi operasi I/O disk dengan melacak data akses file dalam memori, sehingga menghasilkan kecepatan akses yang lebih cepat.

5. Hak akses:

Izin yang dimiliki setiap proses terkait berkas tertentu. Misalnya, suatu proses mungkin memiliki hak baca, tulis, atau eksekusi berdasarkan mode aksesnya. Hak akses menegakkan keamanan dan memastikan bahwa proses hanya dapat berinteraksi dengan berkas sesuai dengan cara yang diizinkan, mencegah akses atau modifikasi yang tidak sah.

d. Tipe File

Salah satu pertimbangan penting dalam merancang sistem file dan keseluruhan sistem operasi adalah apakah sistem operasi mengenali dan mendukung sistem file. Bila sistem operasi mengenali tipe suatu file, maka dapat dilakukan operasi terhadap file dengan lebih baik.

UNIX menggunakan *magic number* yang disimpan pada awal file untuk mengindikasikan tipe file berupa program *executable*, *batch file* (shell script), file *postscript*, dan lain-lain. Tidak semua file mempunyai *magic number*, sehingga informasi tipe tidak dapat digambarkan. *UNIX* tidak menyimpan nama dari program pembuatnya. *UNIX* juga mengizinkan nama ekstensi dari file tersembunyi, sehingga pengguna dapat menentukan tipe file sendiri dan tidak tergantung pada sistem operasi.

File Type	Usual Extension	Function
Executable	exe, com, bin or none	Read to run machine-language program
Object	obj, o	Compiled, machine language, not linked
Source code	c, cc, java, pas, asm, a	Source code in various languages
Batch	bat, sh	Commands to the command interpreter
Text	txt, doc	Textual data, documents
Word processor	wp, tex, rrf, doc	Various word-processor formats
Library	lib, a, so, dll	Libraries of routines for programmers
Print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
Archive	arc, zip, tar	Related files grouped into one file, sometimes compressed, for archiving or storage
Multimedia	mpeg, mov, rm	Binary file containing audio or A/V information

Table 1: File Type