Introduction

Madiba Hudson-Quansah

Contents

Chapter 1

Introduction

Definition 1.0.1: Operating System

Central software component that manages all hardware and software, controlling

- Files, devices, section of main memory, and CPU time
- Who can use the system and how the system is used

Definition 1.0.2: Subsystem

A component of an operating system that manages a specific resource.

Operating systems include for essential subsystem managers, where each manager works with other manager, and performs a unique role.

- Memory Manager
- Processor Manager
- Device Manager
- File Manager
- Optionally, Network Manager

Each manager performs the following classes of tasks:

- Monitor the system's resources continuously
- Enforce the system's security and protection mechanisms

Definition 1.0.3: User Interface

Allows the user to issue commands to the operating system.

1.1 Memory Manager

In charge of main memory (RAM) and read-only memory (ROM). The memory manager is responsible for:

- Checking validity and legality of memory space request
- Reallocating memory to make more useable space available
- Deallocating memory to reclaim it
- Protecting space in main memory occupied by the operating system

1.1.1 ROM

Holds firmware, and is non-volatile.

Definition 1.1.1: Firmware

Determines when and how to load each piece of the operating system after the power is turned on. For example, loading the kernel, BIOS, etc.

1.2 Processor Manager

In charge of the CPU.

- Tracks process status, including which process is currently using the CPU
- Reclaims the CPU when a process if finished or reached the maximum computation time.

Processes are identified by a unique process ID (PID), which is stored in a process control block (PCB).

1.3 Device Manager

In charge of connecting with every available device, responsible for:

- Choosing the most efficient resource allocation method, i.e. scheduling processor.
- Identifying each device uniquely
- Starting device operation when appropriate.
- Monitoring device operation and progress
- Deallocating the device when the operation is complete

1.4 File Manager

In charge of tracking every file in the system, data files, program files, compilers, application programs, etc. is responsible for:

- Enforcing user/program resource access restrictions
- Controlling user/program modification restrictions
- Allocating space for a file on secondary storage
- Retrieving files efficiently

1.5 Network Manager

In change of sharing resources between multiple computers, responsible for:

1.6 User Interface

Definition 1.6.1: User Interface

The section of the operating system that allows fir direct interaction with users

There are two primary types of user interfaces:

- Graphical User Interface (GUI)
  - Input from pointing devices

    - Menu options, desktop metaphors, and windows
- Command-Line Interface (CLI)
    - Input from keyboard
    - Commands with options and arguments

1.7 Types of Operating Systems

Operating Systems fall into five categories:

- Batch
- Interactive
- Real-time
- Hybrid
- Embedded

These categories are distinguished by two features:

- Response time
- Method of data entry

1.7.1 Batch Systems

- Jobs entered as a whole and in sequence
- Input relied no punched cards or tape
- Efficiency measured in throughput

1.7.2 Interactive Systems

- Allows multiple jobs
- Faster turnaround than batch systems
- Slower than real-time systems
- Introduced to provide fast turnaround when debugging programs
- Requires complex algorithms as jobs share processing power

1.7.3 Hybrid Systems

- Combination of batch and interactive systems
- Light interactive load
- Accepts and runs batch programs in the background

1.7.4 Real-time Systems

- Reliability is critical
- Used in time-critical environments (Spacecraft, Airport traffic control, Fly-by-wire aircraft, etc.)
- Two types of real-time systems

  Hard real-time systems - Risk total system failure if a single predict time deadline is missed

  Soft real-time systems - Suffer performance degradation as a consequence if a deadline is missed

1.7.5 Network Systems

- Special class of software
- Users perform tasks using few, if any, local resource, e.g. cloud computing
- Wireless Networking Capability
- Standard feature in many computing devices (cell phone, tables, and other handheld web browsers)

1.7.6 Embedded Systems

- Computers placed inside other products (automobiles, digital music players, elevators, pacemakers, etc.)
- Adds features and capabilities
- Performs a specific set of programs/functions
- Non-interchangeable among systems
- Small kernel and simple function capabilities

Chapter 2

Early Memory Management Systems

2.1 Single-User Contiguous Scheme

- Entire program is loaded into memory
- Contiguous allocation of memory space
- Jobs are processed sequentially
- The memory manager performs minimal work
  - Evaluates incoming process size, loading jobs if small enough to fit in memory

Disadvantages:

- Multiprogramming and networking is not possible, as only one job can be in memory at a time
- Not cost effective, as memory is often idle

2.2 Fixed Partition Scheme

- Memory is divided into fixed number of partitions, where each partition handles one job and reconfiguration requires system shutdown
- This partitioning scheme requires protecting each job's memory space, and matching jobs sizes with partition sizes
- Requires contiguous loading of entire program
- Uses the first available partition with required size method for allocating memory
- To work well all jobs should have similar size and memory size known in advance
- Multiprogramming is possible
- Arbitrary partition sizes can lead to internal fragmentation, i.e. wasted space within a partition

2.3 Dynamic Partition Scheme

- Memory is partitioned dynamically as jobs arrive, i.e. a partition conforms to the size of the job
- Jobs are allocated on a first come, first served basis
- After the first partition sizing and allocation, all subsequent jobs are allocated using those partitions that are free and large enough to hold the job
- First job allocation can lead to external fragmentation, i.e. wasted space between partitions

2.4 Best-Fit and First-Fit Allocation

Two methods for free space allocation

- First-Fit
- Best-Fit

2.4.1 First-Fit

Definition 2.4.1: First-Fit

Free and Busy lists are organized by memory locations

- Jobs are assigned to the first available partition large enough to hold it
- Fast, as it searches from the beginning of memory and stops when a large enough partition is found

Advantage - Faster allocation

Disadvantage - Can lead to many small unusable partitions at the beginning of memory

The memory manager keeps two lists:

- Free memory partitions
- Busy memory partitions

Then compares jobs sizes to the free list, allocating the first partition that is large enough to hold the job. If the entire list is searched and finds no memory block large enough to hold the job, the job is placed into a waiting queue, and the memory manager fetches the next job in the queue.

2.4.2 Best-Fit

Definition 2.4.2: Best-Fit

Free and Busy lists are organized by partition size

- Jobs are assigned to the smallest available partition large enough to hold it
- More efficient use of memory, as it searches the entire list to find the smallest

2.5 Deallocation

Definition 2.5.1: Block

Another word for partition

Definition 2.5.2: Deallocation

Releasing allocated memory space

For a fixed-partition system deallocation is trivial as partition sizes are fixed so the partition's busy flag is set to free.

For a dynamic-partition system, the goal of deallocation is reduce external fragmentation, there are three dynamic partition system cases

- The free block is adjacent to another free block
- The free block is to two other free blocks

2.6 Joining Two Adjacent Free Blocks

The two free blocks are combined into one block and assigned to the free list

2.7 Joining Three Adjacent Free Blocks

Chapter 3

Memory Management Includes Virtual Memory

3.1 Paged Memory Allocation

Definition 3.1.1: Sector

A fixed-length contiguous block of data on a disk

Definition 3.1.2: Page Frame / Frame

A fixed-length contiguous block of data in main memory, this could be parts of a job or an entire job

- Incoming jobs are divided into pages of equal size
- Pages are loaded into page frames in main memory
- In the best case pages, sectors and page frames are the same size, with sizes determined by a disk's sector size
- The memory manager prior to program execution:
    - Determines the number of pages in a program
    - Locates enough empty page frames in main memory
    - Loads all program pages into page frames
- Programs can be stored in non-contiguous page frames
- Internal fragmentation can occur if a page is not completely filled and only happens on the job's last page

There are three tables used to track pages:

- Job Table (JT)
- Page Map Table (PMT)
- Memory Map Table (MMT)

3.1.1 Job Table (JT)

The job table stores information for each active job

- Job Size
- Memory location of the job's PMT

3.1.2 Page Map Table (PMT)

The page map table stores information for each page in a job. Every job has its own PMT, which includes:

- Page number starting from 0
- Memory address of the page frame where the page is loaded

3.1.3 Memory Map Table (MMT)

The memory map table stores information for each page frame in main memory, which includes:

- The locations of the page of which this frame is holding
- Free/Busy status of each frame

3.1.4 Lines

Definition 3.1.3: Line

The smallest unit of data that can be transferred between main memory and the CPU. A page frame is made up of multiple lines.

To determine the page number and displacement of a line we:

1. Divide the job space address by the page size
2. The page number is the integer quotient
3. The displacement is the remainder

To determine the exact location of an instruction or data item in main memory we:

1. Determine the page number/displacement of the line
2. Refer to the job's PMT to determine the page frame containing the required page
3. Obtain the beginning address of the page frame
4. Multiply the page frame number by the page size
5. Add the displacement to the starting address of the page frame

This is also called address resolution / translation converting a logical address (job space address) to a physical address (main memory address).