4 Operating Systems

4.1 What is an Operating System?

[Description of what an os is.]

4.2 User Interfaces

[CLI, menu-driven, GUI]

4.3 An Overview of Different Kinds of Operating Systems

[Batch processing, transaction (interactive) processing, real-time os, multi-tasking (pre-emptive, cooperative) os, time-sharing os, network os, client-server, p2p, distributive system, single-user os, multi-user os, multiprocessing os, multithreading os, online os, multiprogramming os, JCL, job]

4.4 The Main Functions of an Operating System

[the five main functions of an os]

4.4.1 Process Control

[defn of process, this section is about the problems and the solutions of having more than one program executed concurrently by the computer]

4.4.1.1 Process Management

[an example that shows an inefficient use the cpu]

4.4.1.2 CPU Sharing

[pre-emption, context switch, dispatching]

4.4.1.3 States of a Process

[executing, ready (waiting), blocked (suspended), dispatcher]

4.4.1.4 Scheduling

[defn of scheduling, scheduling algorithms FCFS, SJF, priority scheduling, RR, multilevel scheduling]

4.4.1.4.1 First Come First Served (FCFS)

[non pre-emptive]

4.4.1.4.2 Shortest Job First

[non pre-emptive but in pre-emptive mode it is SRTF]

4.4.1.4.3 Priority Scheduling

[starvation, aging, examples of non-pre-emptive and preemptive priority scheduling]

4.4.1.4.4 Round Robin

[time-slice, time-sharing]

4.4.1.4.5 Multilevel Scheduling

[a multi-level queue with different scheduling algorithms within each queue]

4.4.1.5 Process Control Issues

4.4.1.5.1 Deadlock

[defn of deadlock]

4.4.1.5.2 Critical Regions

[defn of critical regions, critical regions are directly related to mutual exclusion]

4.4.1.5.3 Semaphores

[defn of semaphore, lock (wait), unlock (signal), semaphore is directly linked to critical regions]

4.4.1.5.4 Mutual Exclusion

[defn of mutual exclusion]

4.4.1.5.5 Circular Wait

[defn of circular wait]

4.4.1.5.6 Conditions for a deadlock

[3 conditions that may cause deadlock, a fourth condition that will definitely cause deadlock]

4.4.1.5.7 Resource-Allocation Graph

[defn of resource-allocation graph]

4.4.1.5.8 Methods for Handling Deadlocks

[deadlock prevention, deadlock avoidance, deadlock detection and recovery, ignore problem]

4.4.1.5.9 Deadlock Prevention

[Elimination of one of the four conditions for deadlock is discussed]

4.4.1.5.10 Deadlock Avoidance

[real-time techniques that will avoid potential situations that can cause deadlocks]

4.4.1.5.11 Deadlock Detection

[techniques of how to deal with a deadlock when one is detected; recovery from a deadlock; rollback, starvation]

4.4.2 Memory Management

[management of both memory and storage]

4.4.2.1 Storage, Memory and Cache

[brief comparison between the three]

4.4.2.2 Main Memory Characteristics

[addresses, contents]

4.4.2.3 Uniprogramming and Multiprogramming

[one process, many processes]

4.4.2.4 Memory Management Performance

[better performance if there are many processes going on]

4.4.2.5 Memory Management Properties

[kernel space, user space]

4.4.2.6 Relocation

[change logical address to physical address]

4.4.2.7 Protection

[care not to have jumps to other processes, PCB]

4.4.2.8 Sharing

[processes executing the same program,Cooperating processes]

4.4.2.9 Logical and Physical Organisation

[memory is linear, other structures may be non-linear]

4.4.2.10 Methods of Memory Management

4.4.2.10.1 Early Memory Management

[in the old days: processes entirely in RAM, contiguous locations; now: swapping]

4.4.2.10.2 Memory Overlaying

[involves swapping of whole or part of programs, large programs are divided in overlay files by programmers, can cause ‘disk trashing’]

4.4.2.10.3 Partitioning

[partition is a set of contiguous locations in physical memory, designed to hold a single process; fixed partitions, dynamic partitions; fragmentation, placement, replacement, free space management]

Fixed Partitioning

[with equal or unequal sizes, internal and external fragmentation]

Placement Algorithm

[single queue, multiple queues]

Dynamic Partitioning

[size, number of partitions, determined dynamically]

4.4.2.10.4 Segmentation

[memory space is divided into variable sized blocks of contiguous locations called segments, each address is composed of (i) a segment number and (ii) offset, also programs are divided into segments, logical addresses expressed as (s, d)]

An Example of Segmentation

Placement Algorithms for Segmentation

[best fit, worst fit, first fit, next fit]

Free Space Management

[a list, fixed or dynamic]

Fragmentation

[internal fragmentation is not a problem, external fragmentation solved by compaction]

Replacement Algorithms

[which process to replace?]

Address Mapping

[relative address, address translation (address mapping)]

4.4.2.10.5 Paging System

[paging is used for better utilization of main memory, main memory will not necessarily hold whole processes in memory, page frames, pages, virtual memory, real address, virtual address, page fault]

Replacement Algorithms

[random replacement algorithm, FIFO, first-in-not-used-first-out, last recently used]

Direct Mapping

[how a real address is obtained from a virtual address]

Page Size Considerations

[large or small page?]

4.4.3 File Management

[blocks of files and disks]

4.4.3.1 File Organisation and Access

[serial files, sequential, indexed, indexed sequential, direct, or hashed, byte-stream files]

4.4.3.1.1 Serial File

[not sorted]

4.4.3.1.2 Sequential File

[sequential but sorted]

4.4.3.1.3 Indexed Sequential File

[usually used sequentially but has index]

4.4.3.1.4 Direct Files

[indexed, hashed]

4.4.3.1.5 Byte-Stream File Organisation

[treated as a sequence of bytes, pointer]

4.4.3.2 File Directories

[contain lists of files and their attributes]

4.4.3.2.1 Directory Structure

[single-level, two-level directory, tree-structured directory, pathname]

4.4.3.3 File Sharing

[access rights, simultaneous access, read, write, execute, append, delete]

4.4.3.4 File Manipulation

[FMS, buffer]

4.4.3.5 Simultaneous File Access

[mutual exclusion when writing is involved]

4.4.3.6 Blocking

4.4.3.6.1 Blocks on Disk

[Blocking is the process of packing logical records into physical blocks.

4.4.3.6.2 Blocks in Files

[fixed blocking, variable length spanned blocking, variable length un-spanned blocking]

4.4.3.7 Secondary Storage Management

[a file is a collection of blocks, file allocation, free space management]

Allocation Issues

[pre-allocation, dynamic allocation, portion size, data structure to keep track of portions]

Pre-Allocation Versus Dynamic Allocation

[pre-allocation: over-estimation is wasteful]

Portion Size

[variable or fixed-size]

File Allocation Methods

[FAT, 3 allocation methods: contiguous, chained, indexed]

Contiguous Allocation

[simple]

Chained (linked) Allocation

[dynamic allocation, virtually no fragmentation, random processing is made inefficient]

Indexed Allocation

[FAT]

4.4.3.7.1 Free Space Management Issues

[bit vectors/tables, chained free portions, indexing]

4.4.3.8 File System Reliability

[caching, fault tolerance]

File Backup

[full backup, incremental backup, differential backup]

Transaction Logging

[all changes are logged (when, by whom, etc.); high degree of protection, performace penalty]

File Recovery

[backup logs, utility]

Mirroring

[fault tolerance technique, write simultaneously to two disks, high degree of protection]

4.4.4 Handling of I/O Operations

4.4.4.1 I/O Addressing

[isolated/separated I/O, memory-mapped I/O]

4.4.4.2 Handshaking of Devices

[agreement for a communication]

4.4.5 Interrupts

4.4.5.1 What is an Interrupt?

[interrupt handler (ISR), hardware interrupt (asynchronous), IRQ, software interrupt, trap, exception, interrupt-driven system]

4.4.5.2 Polled and Vectored Interrupt

[polled interrupt: interrupt is not specific and interrupt controller must poll devices to know which device sent the interrupt, busy waiting, vectored interrupt includes identity of device sending the interrupt]

4.4.5.3 Interrupt Handler

[associated with each interrupt, interrupt vector table: list of interrupts with associated handlers]

4.4.5.4 System Stack

[stores information about subroutines, overflow]

4.4.5.5 Multiple Interrupts and Interrupt Priorities

[a multiple interrupt is an interrupt that interrupts an interrupt handler. Two techniques to handle multiple interrupts: disable interrupts, interrupt priorities; interrupt latency, interrupt execution time)

Interrupt Mask Register

[makes use of AND operation]

4.4.5.6 DMA

[allows data to be sent directly from an attached device to RAM, DMA's alternative is the PIO]