Chapter 1

Definitions

1.1 What is a Process?

The definition of a process is that, it is the environment a program is executed in. A process is a running program with all of its concomitant data. It is like a virtual machine, BUT with limited acces to e.g. the processor and I/O devices. It is also isolated from other processes.

1.2 What is an Operating System(OS)?

A operating system can be seen as three things, an application manager, a resource manager and an abstraction.
As a application manager its primary task is to run programs, it should maximise the performance of the running programs, that can be executed sequentially or concurrently.
As a resource manager its primary resource is the execution of the CPU. It manages the external devices, like disks, printers, networks and etc. It has support for development, which means that it can provide libraries for common resources, and also has support for advanced features like sharing and IPC.
As an abstraction means that every process has its own machine, isolated from other machines (processes), it supports development with high level APIs/libraries and has a uniform representation of resources.

1.3 What is a Process Image?

A process image is a collection of process related data, which are the PCB(process control block) and the program(which is the executable machine code), the stack and the heap.

1.4 What is a Process Control Block?

A Process control block is for saving the data of each process. The block contains the PC(Program Counter), memory pointers, context data(its registers) and the I/O status. It also has State and Priority for managing the saved processes.(identifier?)

Chapter 2

Processes and Threads

2.1 Keywords

- Definition of process/thread
- Process-/thread-control block
- 7 state process model
- Process creation
- Process image
- Process/thread switching
- Multi-threading
- P-threads

2.2 Learning Goals

- ... can define and explain the concept of a process
- ... can explain what a process image is
- ... can explain what a process control block, what it is used for and why it is needed
- ... can explain, in general terms, how process creation, switching and termination works
- ... can define and discuss process states

2.3 What is a Process?

The definition of a process is that, it is the environment a program is executed in. A process is a running program with all of its concomitant data. It is like a virtual machine, BUT with limited access to e.g. the processor and I/O devices. It is also isolated from other processes.

2.4 What is a Process Image?

A process image is a collection of process related data, which are the PCB(process control block) and the program(which is the executable machine code), the stack and the heap.

2.5 What is a Process Control Block?

A Process control block is for saving the data of each process. The block contains the PC(Program Counter), memory pointers, context data(its registers) and the I/O status. It also has State and Priority for managing the saved processes.(identifier?)

2.6 Creation, switching and termination

A process is created upon login, spawned by other processors or when a new job is started by the OS or User.
A process is terminated when the program is completed, or there is an error condition like a invalid instruction or arithmetic errors, or there is a lack of resources like memory. It can also be terminated from request by its parent or a parent termination.

Switching from processor to processor, is relatively easy. Save the relevant data from the current processor into a PCB, and re-instate the relevant data for the old processor. Relevant data is Status registers like PC(program counter), SP, PSW etc., Data registers like EAX, memory pointers for code and data etc. and the I/O status.
The OS maintains the processors and linking it them with the PCBs. Since a I/O operations takes a lot of time to finish, in the mean time another process can execute. So when do the processor switch back, when the OS regains control, OS regains control when, there is an interrupt like, Clock interrupt I/O, interrupt or memory fault. It can also regain control through system calls or errors(traps).

2.7 Processor states(in PCB)

- Waiting
- Ready
- Suspended
- Ready/Suspended
- Blocked/Suspended

2.8 What is a Thread?

The definition of a thread is an executable part of a process, or the smallest executable part of a processor.

2.9 Multi-Threading

Chapter 3

Concurrency

3.1 Keywords

- Multi-threading
- Implementation strategies for concurrency
- Concurrency versus parallelism
- Inter-process communication
- Race condition
- Mutual exclusion
- Ensuring mutual exclusion
  - Algorithms
  - Hardware supported
  - Mutexes
  - Semaphores
  - Monitors
- Relative time
- P-thread

3.2 Algorithms

Dekker's Algorithm works, efficient when blocking is not needed.
Peterson's Algorithm works, scales to more processors, still generally inefficient, guarantees fairness.

Chapter 4

Deadlocks and Deadlock Handling

4.1 Keywords

- Definition of deadlock
- Mutual exclusion
- Ensuring mutual exclusion
  - Algorithms
  - Hardware supported
- Resource allocation graph
- Coffman's condition
- Solution strategies
  - Prevention
  - Avoidance
  - Detection and recovery
- How to achieve deadlock prevetion (breaking coffmans conditions)
- Safe states and deadlock avoidance
- Banker's algorithm
- Deadlock detection and recovery
- Priority inversion

Chapter 5

Memory Management

5.1 Keywords

- Memory hierarchy
- Challenge in managing memory for an OS
- Physical versus logical versus virtual addresses
- Simple static allocation
- Simple dynamic allocation
- (Simple) Paged memory
- Shared memory
- Implementing paged memory (page tables)

Chapter 6

Memory Management(Virtual Memory, Page Replacement)

6.1 Keywords

- Challenges in memory management
- Virtual memory
- Fetch policy
- Placement policy
- Page replacement policy
- Frame allocation

Chapter 7

File Systems

7.1 Keywords

- Definition of file systems
- Functionality
- Purpose of a file system
- Name space organisation
- Inodes and links
  - Hard
  - Symbolic
- Disk block allocation
- Virtual file systems
- (Harddisk) I/O scheduling

Chapter 8

I/O, System Calls, Device Drivers

8.1 Keywords

- Types of I/O
  - Programmed
  - Interrupt-driven
  - DMA
- Implementation of I/O (as system calls)
- Definition and implementation of system call
- Definition and purpose of device drivers
- Types of device drivers
  - Block
  - Characters
- Device driver implementation
- I/O scheduling (I/O manager)

Chapter 9

Scheduling

9.1 Keywords

- Interrupts
  - Definition
  - Purpose
  - Implementation
- Definition of and goals for scheduling
- Scheduling policies for different systems types
  - Batch
  - Interactive
  - Real-time

Chapter 10

Security

10.1 Keywords

- CIA-triad
  - Definition
  - Explanation
  - Purpose
- OS security
  - Process isolation
  - Strong memory management
  - Permissions
- Application/Software security
  - Common security vulnerabilities in C/Java/C# etc.