Systems Software (1.2.1)

The kernel: Memory Management, Scheduling, Data Security. Memory Management: Gives each process its own segment of memory, keeps track of where each segment is. Stops the segments overwriting each other. Manages Virtual Memory. Device drivers: Allow the OS to control and communicate with peripherals. User interfaces: Allow a person to control and communicate with a computer. Graphical User Interfaces (GUI): WIMPS. Command Line Interfaces: Type written commands.

System utilities: Run in the background; File Repair, Anti-Virus, File Compression. The user only interacts directly with Application programs (through the user interface). The programs use the device drivers to interact with the hardware. The user does not interact with the OS itself.

The BIOS is stored on a BIOS chip containing ROM and Flash memory. When the computer starts, this program begins execution. It carries out hardware and system health checks (POST test). After this, the BIOS will initialise the computer peripherals. Loads the OS.

Segmentation:

- Divides up memory, does not physically divide memory
- Always used; used to unload and load processes.
- Variable sized segments, Each segment is split into four parts, Non-Contiguous

Virtual memory: Where sections of secondary storage act as RAM, when RAM gets too full. Very slow for the CPU to access. When the CPU needs data in VM, the memory manager re-loads it into RAM, swapping out less-used data to make room for it. Paging is used to do this. Pages are swapped, not segments. MM swaps out the least-used page, and swaps in the page you need. Disk Thrashing: If VM is in heavy use, there will be a high rate of data transfer between VM and RAM, causing a high rate of disk access. Prevent Disk Thrashing: Add more RAM, Shut down applications that are not in use.

Paging:

- Divides up memory, does not physically divide memory
- Used to swap data in and out of VM
- Fixed sized pages, Not divided into smaller parts, Contiguous

Scheduling: The task of allocating CPU processor time amongst all running programs. Used by a Multi-tasking OS. Processes as many tasks as possible in a given time. Completes high priority jobs before low priority jobs. Stops Process Starvation. Process Blocking: Where one process enters the running state but does not have the data it needs from the hard disk. It is blocked until it has the data it needs.

Interrupt: A signal. The CPU is running its current program and an interrupt arrives. If the interrupt has a higher priority than the next process, the CPU will suspend its task. The values of each register are placed onto the system stack. The CPU jumps to the related ISR. Once it is complete, the stored values are loaded back into the registers. The CPU continues with the next task. This is called 'context switching'. If the current process has a higher priority, then the interrupt is ignored until the process is complete. Interrupt Service Routine (ISR): The task that the interrupt is asking the CPU to carry out.

Simple to implement, process starvation (pre-emptive), priority, size, number of jobs per time

First Come, First Served:

- Creates a FIFO queue structure. Each process is allowed to run to completion.
- Advantages: Simple to implement.
- Disadvantages: Once a job starts, other jobs undergo process starvation; priority and size are disregarded; non-pre-emptive

Round Robin

- Each process is given an equal slice of CPU time; if not completed afterwards, it is moved to the back of the queue; the next process begins execution
- Advantages: Simple to implement. Good if all processes have similar sizes and priorities. Pre-emptive.

- Disadvantages: priority and size are disregarded; if processing are very different in this way, RR is very inefficient

Shortest Job First

- Queue is ordered by how many FDE cycles are needed to run to completion (estimated)
- When a new process is opened, it gets slotted in at the appropriate point. If it is shorter than any of the queue, the currently-running program is interrupted and the shorter one runs immediately.
- Advantages: Completes maximum number of jobs in a given time; short jobs are not kept waiting
- Disadvantages: Disregards priority; if a long job is nearly completed, it will still be interrupted when a shorter job arrives, thus long jobs may never complete; Estimations could be wrong; non-pre-emptive

Shortest Time Remaining

- Queue is ordered by how many FDE cycles are remaining to run to completion (estimated)
- When a new process is opened, it gets slotted in at the appropriate point. If it is shorter than any of the queue, the currently-running program is interrupted and the shorter one runs immediately.
- Advantages: Completes maximum number of jobs in a given time; short jobs are not kept waiting; Pre-emptive
- Disadvantages: Disregards priority; long processes may never complete if shorter jobs keep coming in

Multi-Level Feedback Queue

- Multiple Queues with different priority levels; each process is given a priority and put in the correct queue; the queues themselves are sorted with an above algorithm
- Queues are completed in order of priority; processes can be reprioritised to prevent process starvation
- Advantages: Sophisticated use of priority and size; pre-emptive
- Disadvantages: Complex to implement. Inefficient with processes of similar priority.

Distributed: Multiple computers combine their processing power across a network to complete a single computer-intensive task. This is a form of parallel processing. Group of computers is called a Cluster.
Embedded: A system has a single, specific purpose, as part of a larger electrical or mechanical system. Run on dedicated hardware so they run with maximum efficiency, little power and little memory.
Multi-Tasking: Several different processes can be running on a single computer at once.
Multi-User: Allows multiple users to log on to the same computer. Presents users with a personalised workspace, maintains privacy between users, monitors processing time usage.
Real Time: Guarantees that all processes will complete execution within a given time frame (its latency), no matter what data, or how much data, is inputted. Extremely reliable.
Batch: Doing the same job repeatedly with different input data. No intervention needed when the initial commands have been set up.

Virtual Machine: An emulation of a computer system. The OS is installed on specialist software. Imitates the hardware that you want to use. There can be many VMs running on a single host. Virtualisation: Process of creating a VM. Intermediate Code: Code designed for a VM.

Non-Programmer Uses

- Run OS that were never intended to be installed on a personal computer (mobile phone OS, Xbox OS). Once the OS is installed, you can run any software that requires that particular OS.
- You can run old applications that are not compatible with new OS
- Open files that may contain malware; VM may be infected, but not the real computer
- However, programs will run less efficiently; you still need proper licences to run the OS and applications; you cannot create a VM that needs more hardware resources than your real computer has

Programmer Uses

- Do not have to rewrite and recompile their programs to run on every possible OS. Write programs to run on a specific type of VM. The VM can be run on any OS (code is portable). No need to change the source code.
- However, the VM may have security vulnerabilities that allow malware to take control of the computer.