## The OSM:

Processor Management

All of the information needed to keep track of a process when switching is kept in a data package called a **process control block**. This typically contains:

- An ID number that identifies the process
- Pointers to the locations in the program and to its data
- Register contents
- States of various flags and switches
- Pointers to the upper and lower bounds of the memory required for the process
- A list of files opened by the process
- The priority of the process
- The status of all I/O devices needed by the process

201

## The OSM:

Processor Management

- Process switching happens without direct user interference, and each process gets enough CPU cycles to accomplish its task in a reasonable amount of time.
- The operating system requires some CPU cycles to perform the **context switching**
- If too many processes are started, most of the CPU cycles will be used to perform the switching and little real work will be done. This is called **thrashing**, and it usually requires some sort of direct user intervention to stop processes and bring order back to the system.

202

## The OSM:
Processor Management

- One way that OS designers reduce the chance of thrashing is by using **threads**. A thread performs all the CPU-intensive work of a normal process but does not require the extensive process control block of a regular process.
  - A process may start many threads or other processes, but a thread cannot start a process.

203

## The OSM:
Processor Management

In a system with two or more CPUs, the OS must divide the workload among the CPUs, trying to balance the demands of the running processes with the available cycles on the different CPUs.

- **An Asymmetric** OS use one CPU for itself and divides application processes among the remaining CPUs.
- **A Symmetric** OS divides itself among the various CPUs, balancing demand versus CPU availability even when the operating system itself is all that's running.

204

## The OSM:

Memory and Storage Management

**Memory and Storage Management**

When an operating system manages the computer's memory, it attempts to accomplish two general goals:

- Each process must have enough memory in which to execute and processes must not interfere with each others memory.
- The different types of memory in the system must be used properly so that each process can run most effectively.

205

## The OSM:

Memory and Storage Management

The first task requires the operating system to set up **memory boundaries** for individual applications.

Applications are loaded into memory in block sizes determined by the operating system.

- If the block size is 2 kilobytes, then every process that is loaded will be given a chunk of memory that is a multiple of 2 kilobytes in size.

206

## The OSM:

Memory and Storage Management

When RAM is scarce, moving information between RAM and hard disk can greatly expand the space available to applications.

- This technique is called **virtual memory management**.

Ranked in order of speed, the types of memory in a computer system are:

- **High-speed cache** - Fast, relatively small amounts of memory accessible to the CPU through fast connections. Cache controllers predict which data the CPU will need and pulls it from main memory.

207

## The OSM:

Memory and Storage Management

- **Main memory**
  - RAM
- **Secondary memory**
  - Disk, Tape.
  - Serves as <u>virtual RAM</u> under the control of the operating system.

The operating system must balance the needs of the various processes with the availability of the different types of memory, moving data in blocks (called **pages**) between available memory as the schedule of processes dictates.

208

## The OSM:

Device Management

**Device Management**

The path between the operating system and practically all hardware not on the computer's motherboard goes through a special program called a **driver**.

- A driver is a translator between the electrical signals of the hardware and the high-level programming language instructions of the operating system and application programs.
- Drivers take data files and translates them into streams of bits for storage devices, or a series of laser pulses in a printer, for example.

209

## The OSM:

Device Management

Because there are such wide differences in the hardware controlled through drivers, there are differences in the way that driver programs function.

- Most are run when the device is required, and function much the same as any other process.
    - The operating system will frequently assigns a high-priority to drivers, so that hardware resources can be used quickly.

210

## The OSM:

Device Management

Drivers are separate from the OS so that the hardware can be changed without changing the OS.

Input and output between different devices and the processor is typically performed using **queues** and **buffers**.

- These store bits from a device for later consumption *en mass* by the CPU and visa versa.

211

## The OSM:

Application Interface

**Application Interface**

Just as drivers provide a way for applications to make use of hardware subsystems, **application program interfaces** (APIs) allow programmers to make use of functions of the computer and operating system

- APIs are very important in the computer industry. Companies often give free access to their software via APIs to encourage programmers to develop applications based on their code – thus giving the company added market share.

212

## The OSM:

Application Interface

**User Interface**

Just as the API provides a structured and consistent way for applications to use the resources of the computer system, a user interface (UI) brings structure and a consistent "look and feel" to the interactions between a user and the computer.

- In the last few decades, almost all development in user interfaces has been in the area of the **graphical user interfaces** (GUIs).
- Multi-modal interfaces are becoming increasingly common in recent years.

213

## The OSM:

Application Interface

Traditionally, Unix used a text-based **shell interface.**

- This is still the preferred mode of interaction among professionals
- There are also graphical user interfaces, such as X-Windows and Gnome, that make Unix and Linux more like Windows and Apple computers from the user perspective.

The user interface is a program or set of programs forming a layer above the operating system.

The core operating-system functions, the management of the computer system, lie in the **kernel** of the operating system.

214

## The OSM

The ties between the operating-system kernel and the user interface, device drivers, utilities and other software define many of the differences in operating systems today, and will further define them in the future.

215