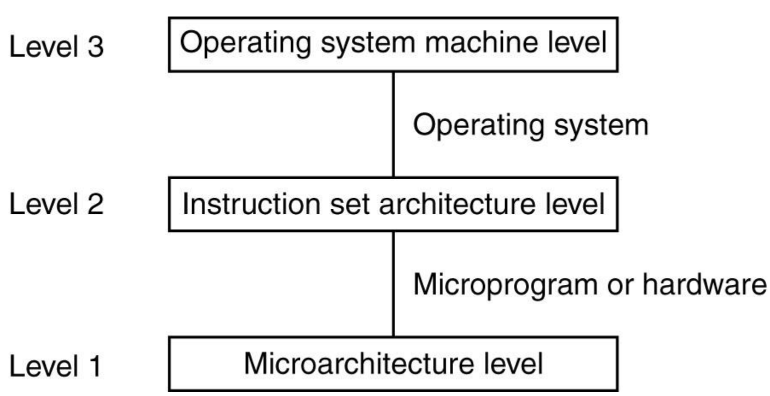
The Operating System Machine Level

186

Operating System Machine



Positioning of the operating system machine level.

187

The OSM

- An **operating system** adds a variety of new instructions and features, above and beyond the ISA level.
- The Operating System level is known as the **OSM** (**Operating System Machine**) level.
- The OSM-level instruction set is the complete set of instructions available to application programmers. It contains nearly all of the ISA level instructions, as
- well as a set of new instructions known as **system calls**.
- A **system call** invokes a predefined operating system service, effectively, one of its instructions. A example system call is to read data from a file.

188

The OSM

The OSM level is always interpreted.

When a user program executes an OSM instruction, such as reading some data from a file, the operating system carries out this instruction step by step.

However, when a program executes an ISA-level instruction, it is carried out directly by the underlying microarchitecture level, without any assistance from the Operating System

Not all computers have Operating Systems.

Many inexpensive microcontrollers do not use an OS.

A microwave oven, for example, has one set of relatively simple tasks to perform, very simple input and output methods (a keypad and an LCD screen), simple, never-changing hardware to control. simply runs a single program all the time.

189

The OSM

At the simplest level, an operating system does two things:

- It manages the hardware and software resources of the computer system. These resources include such things as the processor, memory, disk space, etc.
- It provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware.

190

The OSM

Managing the hardware and software resources

- various programs and input methods compete for the attention of the (CPU) and demand memory, storage and input/output (I/O) bandwidth for their own purposes.
- The operating system makes sure that each application gets the necessary resources

191

The OSM

Providing a consistent application interface

- important if there is to be more than one type of computer using the operating system or if the hardware is to change.
- A consistent application programming interface (**API**) allows a software developer to write an application on one computer and have a high level of confidence that it will run on another computer of the same type, even if the amount of memory or the quantity of storage is different on the two machines.

192

The OSM

- Even if a particular computer is unique, an operating system can ensure that applications continue to run when hardware upgrades and updates occur, because the operating system and not the application is charged with managing the hardware and the distribution of its resources.
- E.g., Windows runs on hardware from thousands of vendors. It can accommodate thousands of different printers, disk drives and special peripherals in any possible combination.

193

The OSM

There are broadly 4 categories of OS:

- **Real-time operating system** (RTOS) - Real-time operating systems are used to control machinery, scientific instruments and industrial systems.
  - A very important part of an RTOS is managing the resources of the computer so that a particular operation executes in precisely the same amount of time every time.

- **Single-user, single task** - designed to manage the computer so that one user can effectively do one thing at a time. The Palm OS for Palm handheld computers is a good example of a modern single-user, single-task operating system.

194

The OSM

- **Single-user, multi-tasking** - This is the type of operating system most people use on their desktop and laptop computers.
  - Windows 98 and the MacOS are both examples of an operating system that will let a single user have several programs in operation at the same time.

- **Multi-user** - A multi-user operating system allows many different users to take advantage of the computer's resources simultaneously.
  - The operating system must make sure that the requirements of the various users are balanced, and that each of the programs they are using has sufficient and separate resources so that a problem with one user doesn't affect the entire community of users.
  - Unix, VMS, and mainframe operating systems

195

The OSM

The operating system's tasks, in the most general sense, fall into six categories:

- **Processor management**
- **Memory management**
- **Device management**
- **Storage management**
- **Application interface**
- **User interface**

196

The OSM

Processor Management

- Two basic tasks:
  - Ensuring that each process and application receives enough of the processor's time to function properly
  - Using as many processor cycles for real work as is possible

The basic unit of work is the **process** or **thread.**

- An application may be composed of many processes.

197

The OSM

- In a single-tasking system, the schedule is straightforward. The operating system allows the application to begin running, suspending the execution only long enough to deal with interrupts and user input.
- **Interrupts** are special signals sent by hardware or software to the CPU looking for the its attention.
- Sometimes the operating system will schedule the priority of processes so that interrupts are **masked**, that is, the operating system will ignore the interrupts from some sources so that a particular job can be finished as quickly as possible.

198

The OSM

- There are some interrupts (such as those from error conditions or problems with memory) that are so important that they can't be ignored. These **non-maskable** interrupts (NMIs) must be dealt with immediately, regardless of the other tasks at hand.

The job of the operating system becomes much more complicated in a multi-tasking system.

- It must share the CPU among many users so that each appears to have full control of the machine.
- The operating system switches between different processes thousands of times a second.

199

The OSM

- A process occupies a certain amount of RAM. It also makes use of registers, stacks and queues within the CPU and operating-system memory space.
- The operating system allocates a certain number of CPU execution cycles to each process.
- After that number has expired, the OS halts the process and save all registers, stacks and queues belonging to that process to memory.
- It then starts a waiting process after loading all its associated registers, stacks and queues

200

The OSM

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