Computer System Software Course
CE 403

Dr. Mohamed Abd El-Latif Mowad
Memory Allocation, Paging, Segmentation

Dr. Mohamed Abd El-Latif Mowad
Email: Turbocode_2000@yahoo.com
Cell phone :01007532006
• In many times we have multiple processes which must **share memory** leading to which is known as **Memory Management**, which is done using O-S.

• Selection of memory management method for a specific system depends on many factors, especially on the system **hardware design**.

• We are interested only in the sequence of memory addresses generating by the running program.
Basic Hardware:

- **Main memory and registers** built into the processor itself are the only storage that the processor can access **directly**.
- Any instruction in execution and any data being used by the instructions must be in one of these direct access storage devices.
- Most processors can decode instructions at the rate of one or more operations **per clock**.
- After the instruction has been executed, results may be **stored back** in physical memory.
- Memory access may take **many cycles** of the processor clock, data required to the processor to complete the instruction is not available so;
• The processor normally needs to **stall**.
• The solution is to **add a fast memory** between the processor and main memory which is called **cache memory**.
• We are not interested only in the **relative speed** of accessing physical memory but we must ensure **correct operation** has to **protect O-S** from access by the used processes, and also **protect users processes** from each others.
• We need to make sure that **each process** has a separate **Memory Space**. To do this we need to determine the range of **legal addresses** that the process can access. We can provide this protection by using **two registers**, usually
Base and Limit as shown in the following figure.

Base and Limit Registers Define a Logical Address space
• The **Base Register** holds the smallest legal physical memory address, the **Limit Register** specifies the size of the range.

• *Ex:* if base register holds 30004
• Limit register is 12090

• It means the **program** can legally access all addresses from 30004 through 42094.

• **Protection** of memory space is done by comparing every address generated in the user mode with the registers. Any attempt by a program executing in **user mode** to access **O-S memory** or other user’s memory results in a **trap** in O-S and **provides a fatal mistake**. This is shown in figure.
Hardware Address Protection with Base Limit Registers
• The base and limit registers can be loaded only by O-S which uses a special instructions executed only in kernel Mode.

• **Logical Address versus Physical Address:**
  • The address generated by the CPU is commonly referred as **Logical Address** *(virtual address).*
  • Memory address register is referred as **Physical Address.**
  • The set of all logical addresses generated by the program is a **Logical Addresses Space.**
  • The set of all physical addresses corresponding to these logical address is a **Physical Address Space.**
• The run time from logical to physical address is done by a hardware device called the **Memory Management Unit (MMU)**.
• The **base register** is called now **relocation register**.

**EX:**
• If the base is at **14000** then the attempt by the user to address location zero is **14000** when access to location **346** is mapped to location **14346**.

• **Dynamic Loading:**
  To obtain better memory space utilization we can use **Dynamic Loading** where a routine is not loading until it is called.
Swapping

A process can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution.

Backing store – fast disk large enough to accommodate copies of all memory images for all users; must provide direct access to these memory images.

Roll out, roll in – swapping variant used for priority-based scheduling algorithms; lower-priority process is swapped out so higher-priority process can be loaded and executed.

Major part of swap time is transfer time; total transfer time is directly proportional to the amount of memory swapped.

System maintains a ready queue of ready-to-run processes which have memory images on disk.
Swapping Process
All routines are kept on disk in reloctable load format. The main program is loaded into memory and is executed. When a routine needs to call another routine, it first checks to see whether the other routine is loaded or not, if not the reloctable linking loader is called to load the required routine into memory.
The advantages of dynamic loading is that unused routine is never loaded.

This method is particularly useful when large amounts of code are needed to handle infrequently occurring cases such as error routines. In this case although the total program size may be large, the portion that is used (and hence loaded) may be much smaller.
The **Base** register is equivalent to **relocation register**.

Hardware support for relocation and limit register.