Unit: III
Lecture: 3
(Memory Management)
Address Binding: Mapping address space to memory space

The Process Address Space is the set of logical addresses that a process references in its code. The operating system takes care of mapping the logical addresses to physical addresses at the time of memory allocation to the program.

There are three types of addresses used in a program before and after memory is allocated:

- **Symbolic addresses:** the addresses used in source code. The variable names, constants and instruction labels are the basic elements of the symbolic address space.
- **Relative addresses:** at the time of compilation, a compiler converts symbolic addresses into relative addresses.
- **Physical addresses:** the loader generates these addresses at the time when a program is loaded into main memory.

Address binding relates to how the code of a program is stored in memory. Programs are written in human-readable text, following a series of rules set up by the structural requirements of the programming language and using keywords that are interpreted into actions by the computer's Central Processing Unit. The point at which the executable version of a program is created dictates when address binding occurs.

Computer memory uses both logical addresses and physical addresses.

Address binding allocates a physical memory location to a logical pointer by associating a physical address to a logical address (virtual address).

Address binding is part of computer memory management and is performed by the operating system on behalf of the application that need access to memory.

Address binding of instructions and data to memory addresses can happen at three different stages:

- **Compile Time:**
  - If you know that during compile time, where process will reside in memory then an absolute address is generated i.e. physical address is embedded to the executable of the program during compilation.

- **Load Time:**
  - If it not known at the compile time where the process will reside then a relocatable address will be generated.
  - The loader translates the relocatable address to an absolute address.
  - The base address of the process in main memory is added to all logical addresses by the loader to generate an absolute address.
- **Execution Time:**
  - The instructions are in memory and are being processed by the CPU.
  - Additional memory may be allocated and/or deallcoated at this time.
  - This is used if a process can be moved from one memory to another during execution.

Virtual and physical addresses are the same in compile-time and load-time address-binding schemes while differ in execution time address-binding scheme.

Logical and Physical Address in Operating System:

Logical Address:

- Logical address is generated by CPU while a program is running.
- The logical address is virtual address as it does not exist physically → it is also known as Virtual Address.
- This address is used as a reference to access the physical memory location by CPU.
- The hardware device called Memory-Management Unit is used for mapping logical address to its corresponding physical address.

Physical Address:

- Physical address identifies a physical location of required data in a memory.
- The user never directly deals with the physical address but can access by its corresponding logical address.
- The user program generates the logical address and thinks that the program is running in this logical address but the program needs physical memory for its execution, therefore the logical address must be mapped to the physical address by MMU before they are used.

Mapping Virtual Addresses to Physical Addresses:

The run-time mapping from virtual to physical addresses is done by a hardware device called the memory-management unit (MMU).

On basic and simple MMU scheme to map virtual address to physical address is a generalization of the base-register scheme.

The base register is now called a relocation register.

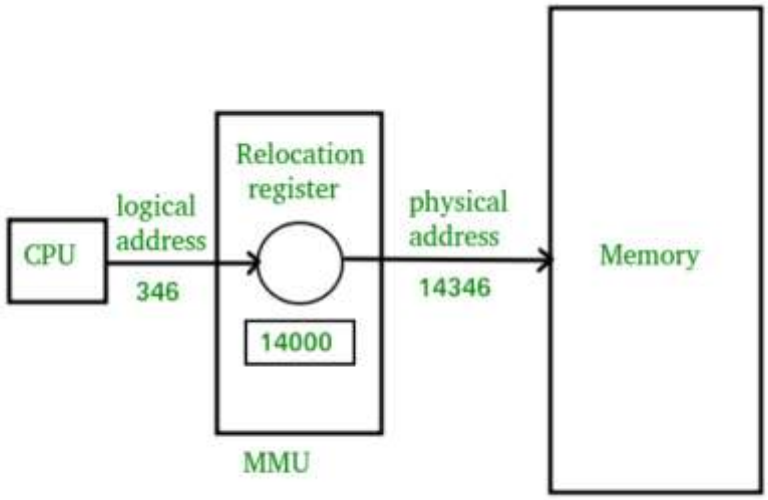

Fig: Dynamic Relocation using a relocation register

The value of the relocation register is added to every address generated by a user process at the time the address is sent to memory.

e.g. if the base is at 1400, then an attempt by the user to address location 0 is dynamically relocated to location 14000.

Similarly an access to location 346 is mapped to location 14346.

The user program never sees the real physical addresses. The program can create a pointer to location 346, store it in memory, manipulate it and compare it with other addresses—all as the number 346.

Only when it is used as a memory address is it relocated relative to the base register.

This scheme is mostly used in contiguous memory allocation.

In non contiguous memory allocation, processes can be allocated anywhere in available space. There are several techniques used for address translation in non contiguous memory allocation like Paging, Segmentation etc.