Real Memory Organization ch7

Memory Management Strategies

-These strategies are designed to obtain the best possible use of main memory. They are divided into:

1. Fetch strategies: determine when to move the next piece of a program or data to main memory from secondary storage. We divide them into demand and anticipatory. In demand fetch strategy, the system places the next piece of program or data in main memory when a running program references it. Today, many systems have increased performance by employing anticipatory fetch strategies, which attempt to load a piece of program or data into memory before it is referenced.
2. Placement strategies: determine where in main memory the system should place incoming program or data pieces, first fit, best fit, and worst fit, memory placement strategies.
3. Replacement strategies: when memory is too full to accommodate a new program, the system must remove some (or all) of a program or data that currently resides in memory. The system's replacement strategy determines which piece to remove.

Contiguous Vs Noncontiguous Memory Allocation

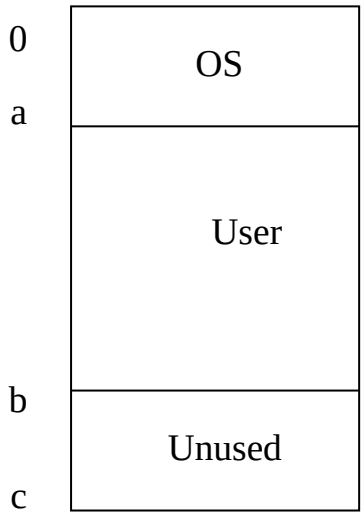
-Contiguous Memory Allocation: to execute a program in early computer systems, the system operator or the operating system had to find enough contiguous main memory to accommodate the entire program. If the program was larger than the available memory then the system could not execute it.

-In Non-Contiguous Memory Allocation: a program is divided into blocks or segments that the system may place in non adjacent slots in main memory. This
allows making use of holes (unused gaps) in memory that would be too small to hold whole programs.

Single-User Contiguous Memory Allocation

-Early computer systems allowed only one person at a time to use a machine. All the machine's resources were dedicated to that user and the user was charged for all the resources whether or not the user's job required them.

-The programmer wrote all the code necessary to implement a particular application including I/O instructions. The designers consolidated (combined) I/O coding that implemented basic functions into an I/O control system (IOCS). The programmer called IOCS routines to do the work.

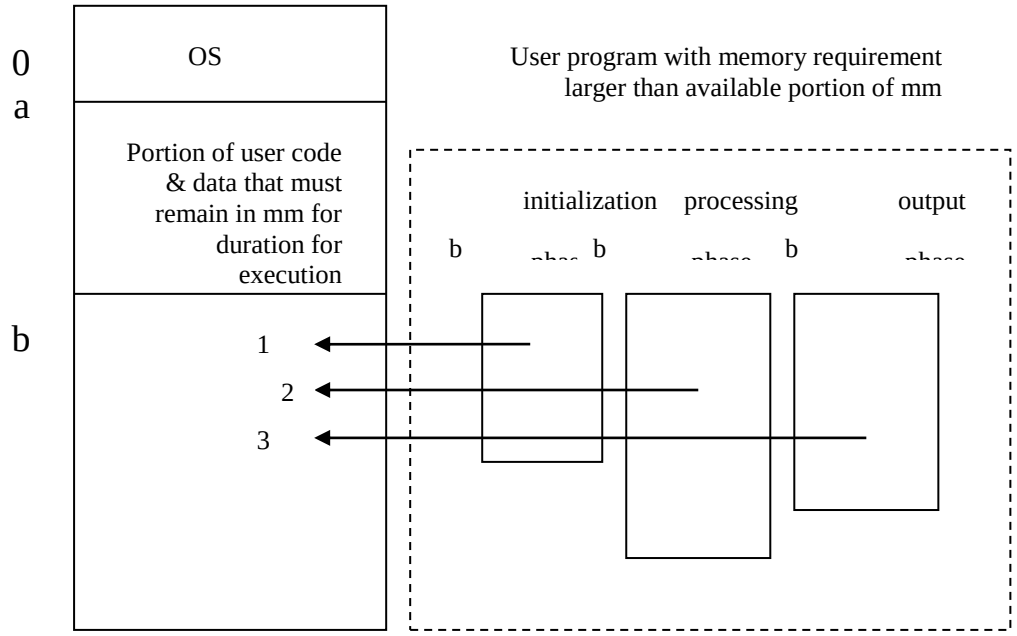


Single-user
contiguous

Overlays

-One way in which a software designer could overcome the memory limitations was to create overlays, which allowed the system to execute programs larger than main memory.

-The programmer divides the program into logical sections. When the program does not need the memory for one section, the system can replace some or all of it with the memory for a needed section. Overlays enable the programmers to extend main memory.



1 load initialization phase at b and run
2 then load processing phase at b and run
3 then load output phase at b and run