Memory Management Systems

Memory Management System

Memory Cycle:

The entire activities (at the circuit level) that takes place to read from or write into a memory space is called a memory cycle. Therefore a memory cycle can be a memory read cycle or a memory write cycle.

Two registers are involved in reading and writing process

1- Memory-Address register (MAR):

Carries the address of a memory space:

from which the data is read by a read operation or

into which the data is written by a write operation

2- Memory-buffer register (MBR):

Carries data (as much as one word) that:

is read from the [MAR] or

will be written into [MAR]

Memory Management System

Logical Address

The address generated by CPU (i.e. the offset address from the base of the executable module within the main memory.)

Physical address

The absolute address of a memory space. (MAR carries the physical address)

Memory management Unit (MMU)

A hardware device in charge of mapping logical address into the physical address.

Logical Memory

Memory space needed for a process

Physical Memory

Main Memory (RAM)

Memory Management System

Bindings

In source program, addresses are symbolic: *Tax = 0.7;*

Compiler binds the symbolic addresses to relocatable addresses
(e.g 22 bytes from the beginning of the object module)

Linker or loader (depends on the system) binds the relocatable address to the absolute address.
(e.g 8950AB)

Memory Management System

Systems

1- Bare machine scheme:

No memory management system at all.

Memory Management System

2- Swapping scheme

Moving a process to a baking store and bring it back to main memory at a later time for continued execution. This method has been exercised, by medium schedulers, round robin, and other priority-based algorithms

Memory Management System

3- Contiguous fixed partition allocation (MFT)

Memory is divided into several partitions. One partition is always reserved for the kernel which is either the partition in low memory or high memory.

A partition must be big enough to accommodate any process.

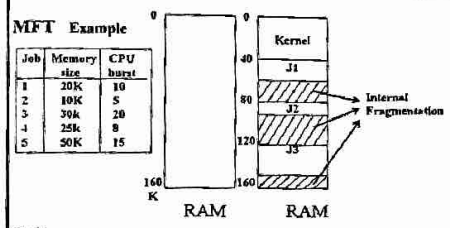
Waste of memory because of internal fragmentation.

(MFT): Multi-programming with a Fixed number of Tasks

Memory Management System

MFT Example

Job	Memory size	CPU burst
1	20K	10
2	10K	5
3	30k	20
4	25k	8
5	50K	15

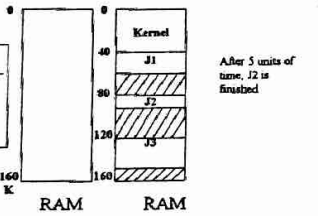


Q = 14

Memory Management System

MFT Example

Job	Memory size	CPU burst
1	20K	10
2	10K	5
3	30k	20
4	25k	8
5	50K	15



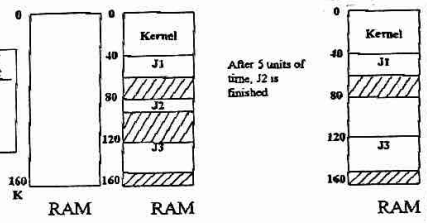
After 5 units of time, J2 is finished

Q = 14

Memory Management System

MFT Example

Job	Memory size	CPU burst
1	20K	10
2	10K	5
3	30k	20
4	25k	8
5	50K	15



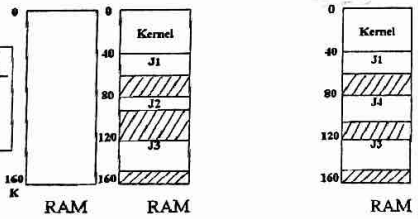
After 5 units of time, J2 is finished

Q = 14

Memory Management System

MFT Example

Job	Memory size	CPU burst
1	20K	10
2	10K	5
3	30k	20
4	25k	8
5	50K	15



Q = 14

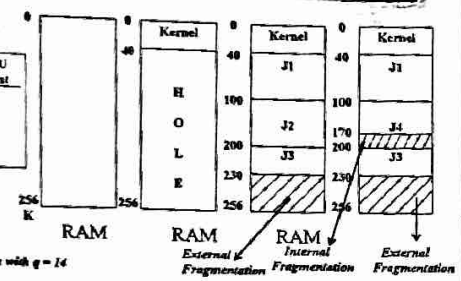
Memory Management System

4- Contiguous variable partition allocation (MVT)

Memory Management System

MVT Example

Job	Memory size	CPU burst
1	60K	10
2	100K	5
3	30k	20
4	70k	8
5	50K	15



Using Round Robin with q = 14

Memory Management System

- The amount of memory allocated to a process is not limited to a pre-determined partition.
- As much space a process needs is given to it from the hole. As soon as a process is finished its space becomes available (becomes a hole) and it will be merged with the adjacent holes, if any.
- As a result, at any given time looking at the original partition (hole) it is the home of a number of processes and a number of smaller holes of different sizes.

Memory Management System

Allocation of space to a new process may use one of the following approaches:

- First Fit
- Best Fit
- Worst Fit

Problem: Fragmentation
- Internal and
- External

Solution: Use of Non-contiguous physical addresses

Memory Management System

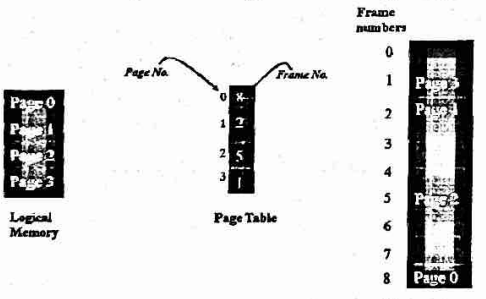
5- Paging

(reminder)

Logical Memory: Memory space needed for a process

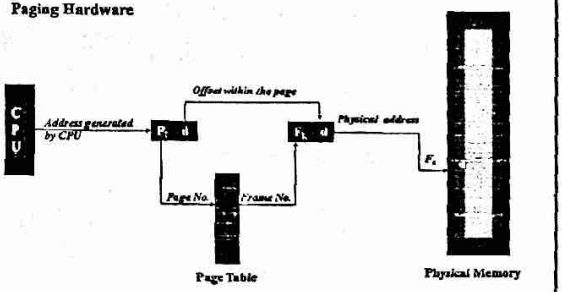
Physical Memory: Main Memory

Memory Management System



Memory Management System

Paging Hardware



Memory Management System

Page Table Structure

1. *Using Registers*
2. *Using Main Memory*
3. *Using Fast Memory (e.g. Cache memory)*

This fast memory has an access time higher than registers but lower than Main memory

Memory Management System

Page Table Structure

- *Using Registers (the entire page table is stored in registers)*
  - *Advantage*
    - *Fast*
  - *Disadvantage*
    - *Not good for large page tables*

Memory Management System

Page Table Structure

- *Using Main Memory (the entire page table is stored in RAM)*
  - ***Advantage***
    - ***Good for large page tables***
  - ***Disadvantage***
    - ***Slow by factor of 2 (Access time is high)***

Memory Management System

Page Table Structure

- *A note about using Fast Memory (e.g. Cache memory).*
- *This Cache memory is different from CPU Cache memory and it is a separate one and it is called* ***Translation look-aside buffer (TLB).***
- *The most referenced (Page#, Frame#) pairs are kept in TLB and the rest are kept in RAM.)*

Memory Management System

Page Table Structure

Using Fast Memory.

- *Advantage*
  - *It improves the access time to "Effective access time" for large page tables*
- *Disadvantage*
  - *Slower than using registers*

Memory Management System

TLB Use