Unit-2

Shalini

Memory management:

An OS is concerned with management of primary memory.
Primary memory means the memory that the processors directly access for instructions and data.

Memory mgmt functions:

- Keeping track of the status of each location of memory (allocated/ de allocated)
- Determine allocation policy for memory i.e. deciding when, where, how much, and to whom.
- Allocation technique-once it is decided to allocate memory, the specific location must be selected and allocated using allocation techniques.
- De allocation techniques-once the execution is over, the process may explicitly release the previously allocated memory or memory management may reclaim the allocated memory using de allocation techniques.

Memory mgmt techniques:

1. Single contiguous memory mgmt.
2. Partitioned memory mgmt.
3. Relocated partitioned memory mgmt.
4. Paged memory mgmt.
5. Demand paged memory mgmt.
6. Segmented memory mgmt.
7. Segmented and demand paged memory mgmt.
8. Other memory mgmt.

1. SINGLE CONTIGUOUS ALLOCATION:

- It is a simple memory mgmt scheme which requires no hardware features.
- It is associated with small stand-alone computers with simple batch operating system.
- There is no multiprogramming & one-one correspondence between a user, a job, & a process.

Main Memory

←Operating system

Allocated for user's job →

←Allocated but unused (wasted)

- In this scheme, memory is divided into 3 contiguous regions.
- A portion of memory is permanently allocated to the OS. Remainder of the memory is available (allocated) to single job being processed.
- The job actually uses portion of the allocated memory, leaving an allocated but unused region of memory.

2

- Eg: if there are 256K bytes of memory, an OS may require 32KB, remaining 224KB allocated for user job. If a job requires only 64KB then remaining 160KB of memory is unused. This is depicted diagrammatically in the above figure.

Functions of single contiguous memory mgmt:

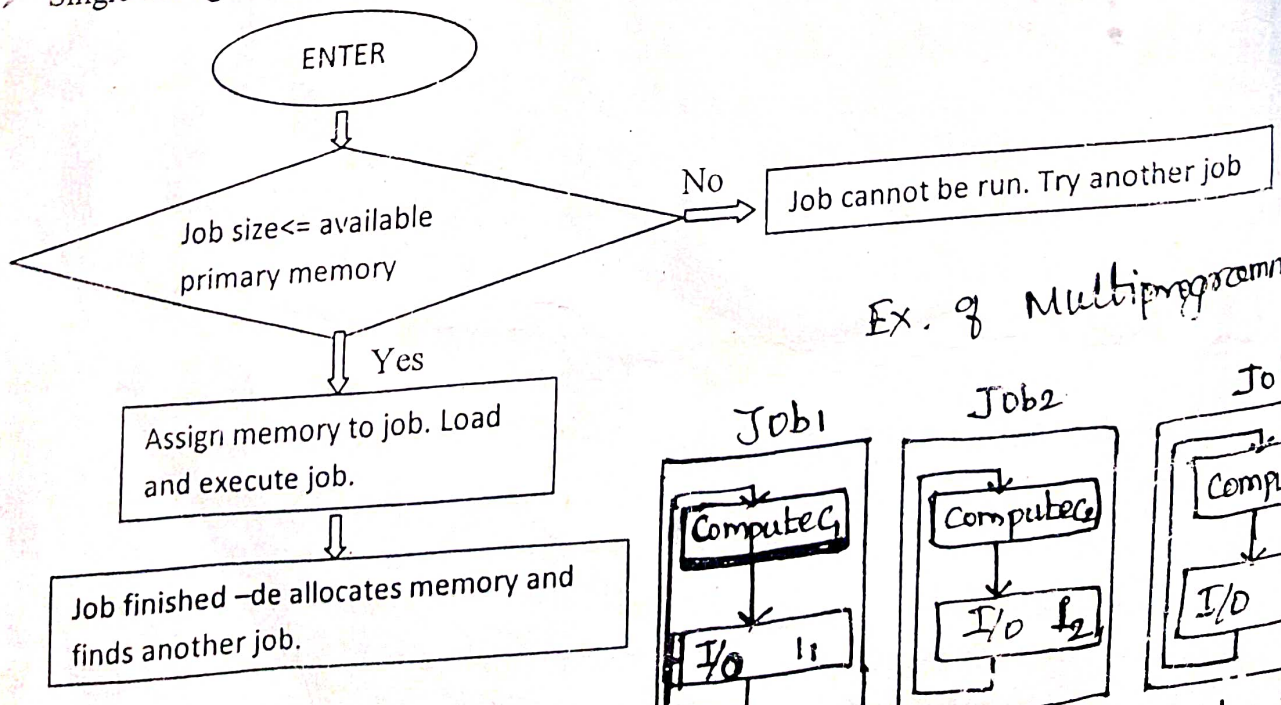
- Keeping track of memory – it is allocated entirely to one job..
- Determining factor on memory policy – the job gets all memory when scheduled.
- Allocation of memory – all of it is allocated to the job.
- De allocation of memory – when the job is done, all memory is returned to free status.

Hardware support:

- No special hardware is required for contiguous allocation.
- Sometimes a primitive hardware protection mechanism is desirable to ensure that the user's programs do not accidentally damage the operating system.
- This mechanism may consist of a bounds register and a supervisor – user mode of the CPU.
- The bound register contains the address of the protected area (which includes OS).
- If the CPU is in user mode, on each reference to memory the hardware checks to assure that it is not an access to the protected area.
- In supervisor mode the OS can access the protected area as well as execute privileged instructions that change the contents of the bounds register.

Software support:

- Single contiguous allocation algorithm is depicted in flowchart as follows



Ex. of Multiprogramming

- Eg: if there are 256K bytes of memory, an OS may require 32KB, remaining 224KB allocated for user job. If a job requires only 64KB then remaining 160KB of memory is unused. This is depicted diagrammatically in the above figure.

Functions of single contiguous memory mgmt:

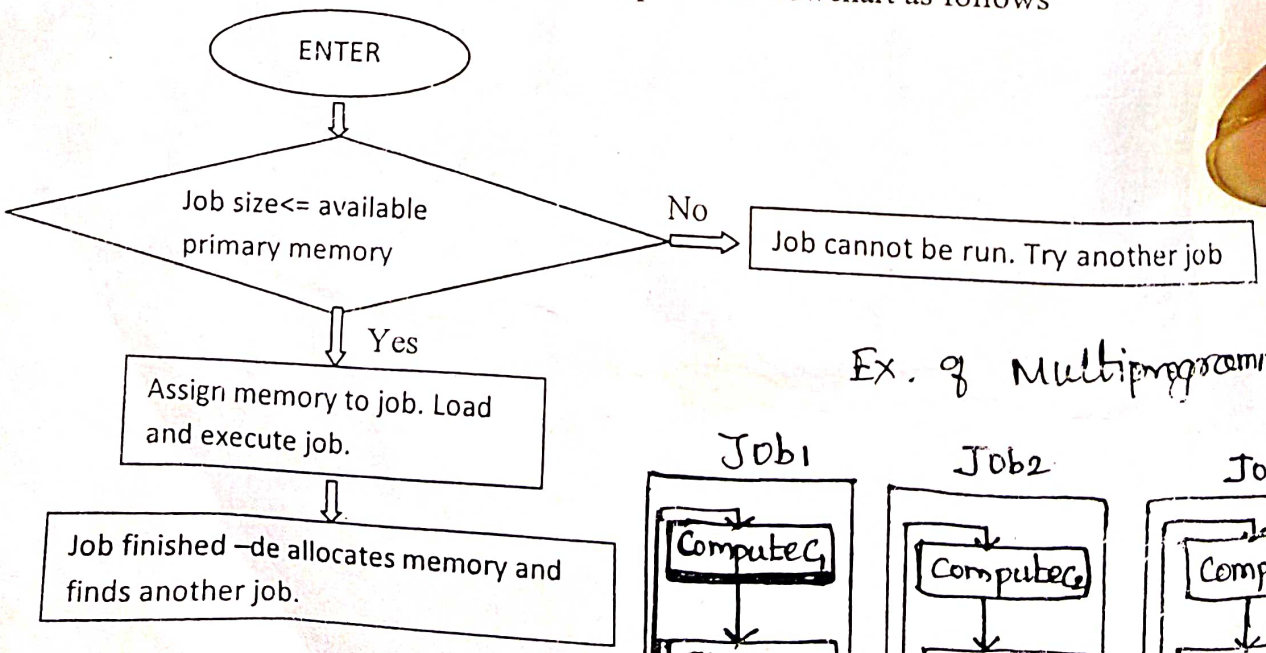
- Keeping track of memory – it is allocated entirely to one job..
- Determining factor on memory policy – the job gets all memory when scheduled.
- Allocation of memory – all of it is allocated to the job.
- De allocation of memory – when the job is done, all memory is returned to free status.

Hardware support:

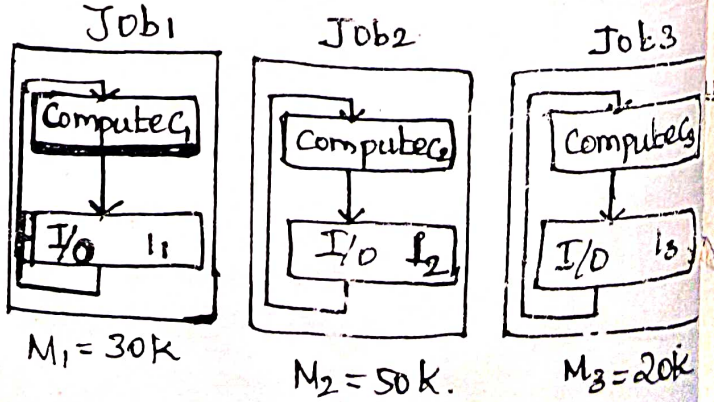
- No special hardware is required for contiguous allocation.
- Sometimes a primitive hardware protection mechanism is desirable to ensure that the user's programs do not accidentally damage the operating system.
- This mechanism may consist of a bounds register and a supervisor – user mode of the CPU.
- The bound register contains the address of the protected area (which includes OS).
- If the CPU is in user mode, on each reference to memory the hardware checks to assure that it is not an access to the protected area.
- In supervisor mode the OS can access the protected area as well as execute privileged instructions that change the contents of the bounds register.

Software support:

- Single contiguous allocation algorithm is depicted in flowchart as follows



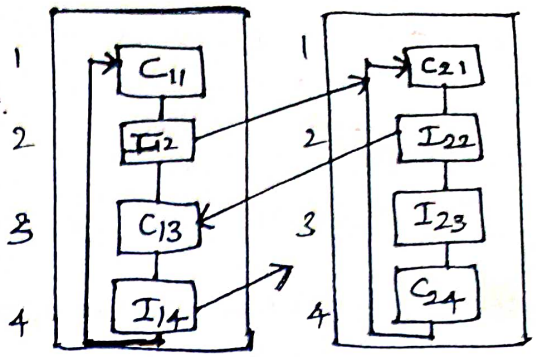
Ex. of Multiprogramming.



- This algorithm is called, when the job scheduler of the processor management wishes to schedule a job to be run.
- The algorithm is called only when no other job is using memory.

Advantages:

- In this scheme, OS uses minimum amount of memory.
- Easy to understand and use.
- It does not require extra hardware.



Disadvantages:

- Poor utilization of memory.
- Poor utilization of processors. (Waiting for I/O)
- User's job being limited to the size of available main memory.
- There is no multiprogramming.

I/O waiting state.

2. PARTITIONED MEMORY MGMT:

- It supports multiprogramming.
- Main memory is divided into separate regions are called memory partition.
- Each partition holds separate job.

Main Memory

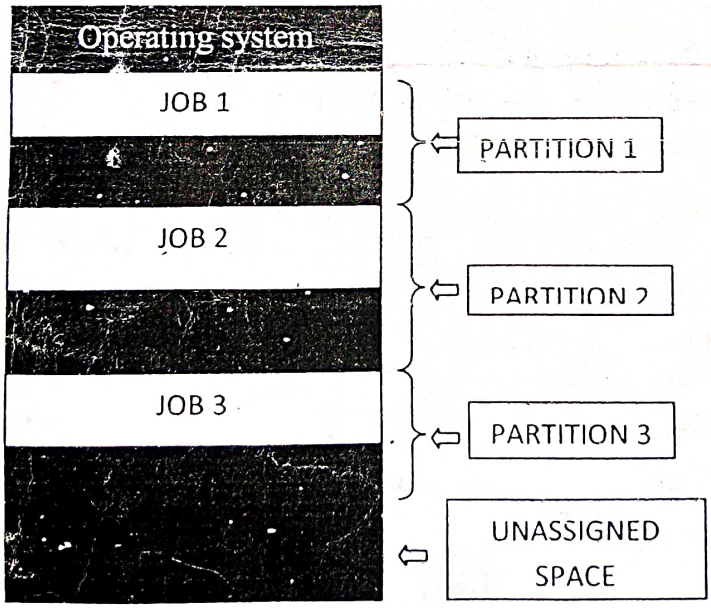


Figure PARTITIONED ALLOCATION

Functions of partitioned memory mgmt:

- Keeping track of the status of each partition. (eg. "in use" or "not in use", size)
- Determining who gets memory-decided by job scheduler.
- Allocation technique-if sufficient partition is available the jobs are allocated.

- De allocation - when the job terminates, the partition is "indicated" not in use" and is available for further allocation.

H /w support:

- Very little special h/w is needed.
- Memory protection mechanism is desirable to prevent from disturbing either OS or other job accidentally
- We could use two bounds register to bracket the partition being used.
- If the job tried to access memory outside the partition, a protection interrupt would occur.
- there are two disadvantages
  - The bounds registers must be changed every time that the processor is reassigned for multiprogramming
  - It is difficult to extend this protection to I/O channel. A job in partition 1 may try to instruct a channel to read data into an area in partition 3. We could provide bounds registers for every I/O channel, with multiple registers required for multiplexor channels.

S /w algorithm:

- The partition allocation approach has two common versions
  a) Static partitioned specification.
  b) Dynamic partitioned specification.

a. Static partitioned specification:

- It is also called fixed partitioned specification. Main memory is divided into number of partitions at before execution of user programs.
- An example of static partition specification table is given below

Partition number	Partition size	location	Status
1	8K	312K	In use
2	32K	320K	In use
3	32K	352K	Not use
4	120K	384K	Not use
5	520K	504K	In use

Static partition specification table

- Each job step supplied by a user must specify the maximum amount of memory needed.
- A partition of sufficient size is then found and assigned.
- This technique is appropriate when the sizes and frequency of jobs are well known.
- In such a case, partition sizes are chosen to the most common job sizes.
- However there are considerable memory wasted if the sizes and frequencies of jobs are not known.

- For eg, if 5 jobs of sizes 1K,9K,9K,33Kand121K are to be run, we can assign these to partitions as follows

Partition number	Partition size	Job size	Wasted space
1	8K	1K	7K
2	32K	9K	23K
3	32K	9K	23K
4	120K	33K	87K
5	520K	121K	399K
TOTAL	712k	173k	539k

- In this case, all the partitions are assigned in the best possible way, yet only 173K of the available 712K is actually used. Hence 75% of the available memory is wasted.

b. Dynamic partitioned specification:

- Dynamic partitioned specification means partitions are created during job processing.
- Tables must be made up with entries for each free area and each allocated partition.
- Two separate tables are used, one for the allocated areas and another for maintaining the status of the unallocated/free areas.

DYNAMIC PARTITION SPECIFICATION TABLES

PARTITION NUMBER (PROTECTION KEY)	SIZE	LOCATION	STATUS
1	8K	312K	ALLOCATED
2	32K	320K	ALLOCATED
3	-	-	EMPTY ENTRY
4	120K	384K	ALLOCATED
5	-	-	EMPTY ENTRY
	-----	-------	---------

TABLE 1- ALLOCATION PARTITION STATUS TABLE

FREE AREA	SIZE	LOCATION	STATUS
1	32K	312K	AVAILABLE
2	32K	320K	AVAILABLE
3	-	-	EMPTY ENTRY
4	-	-	EMPTY ENTRY
5	-	-	EMPTY ENTRY
	-----	-------	--------

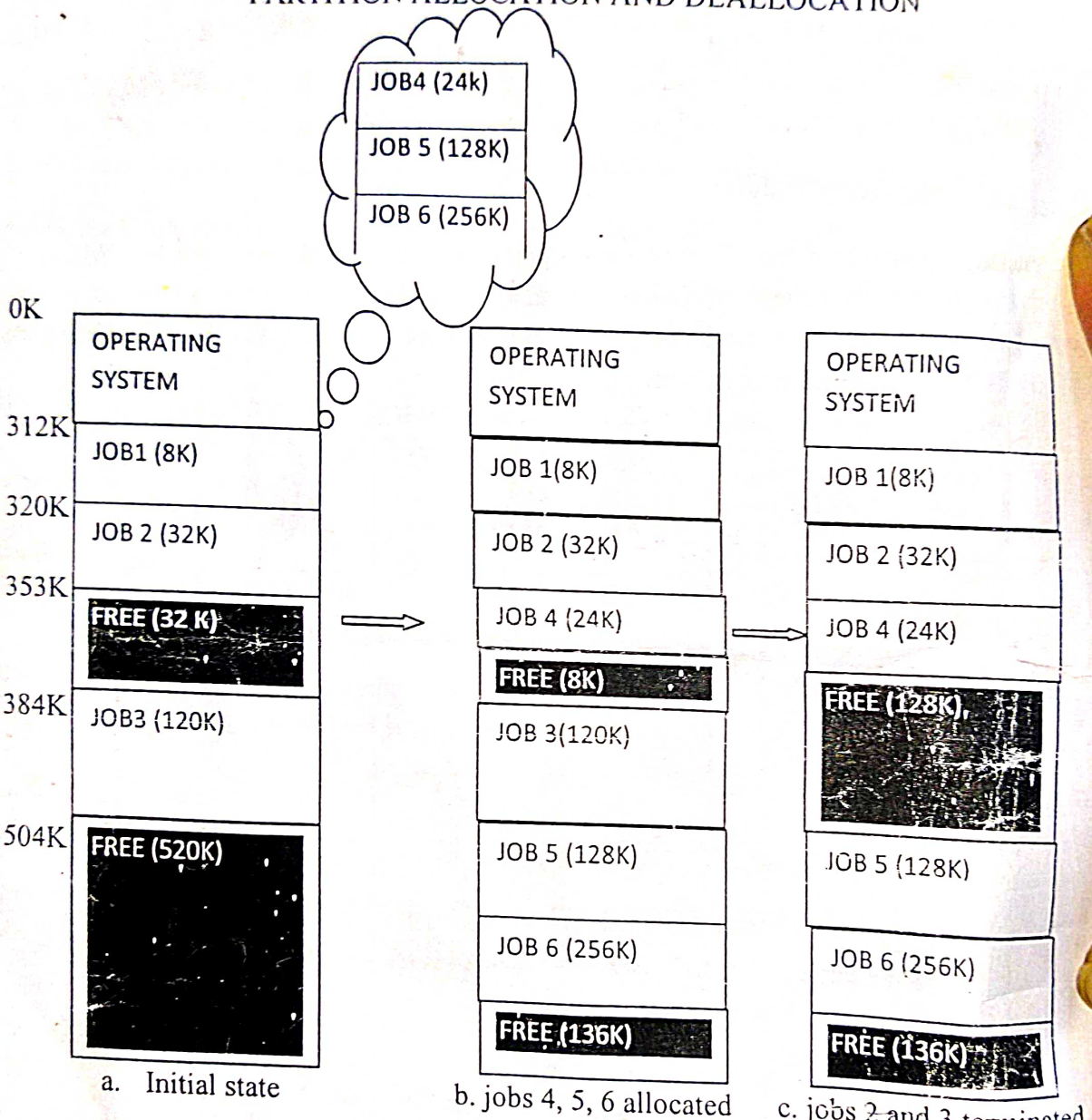
TABLE 2- UNALLOCATED AREA STATUS TABLE

b

Example:

An example of dynamic partition specification is presented in following figures. Three partitions are allocated, each containing a job of corresponding size (fig a). 3 additional jobs are then selected to be multi-programmed and new partitions of appropriate sizes are created from the unallocated free areas (fig b). The partitions can be deallocated after corresponding job is terminated (fig c).

PARTITION ALLOCATION AND DEALLOCATION



a. Initial state b. jobs 4, 5, 6 allocated c. jobs 2 and 3 terminated

For allocation:

i. Free area as large as partition desired must be found.

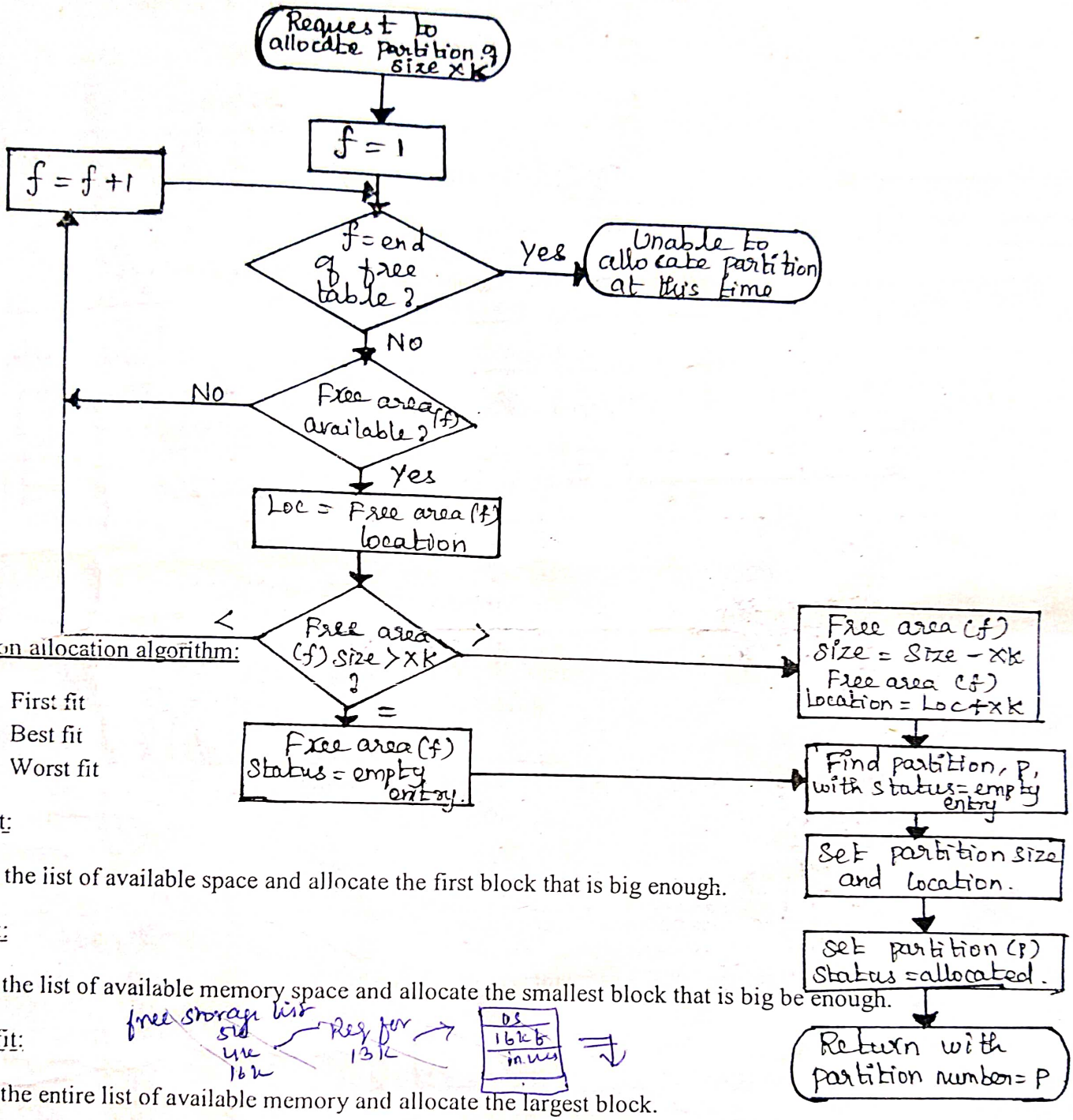
ii. If area is larger than needed, it must be split into two pieces – one becomes partition area and other becomes free area.

For de allocation:

i. Adjacent Free areas should be merged so as to make it as contiguous free area.

S/w algorithm:

- Partition allocation algorithm.
- Partition de allocation algorithm.



artition allocation algorithm:

a) First fit
b) Best fit
c) Worst fit

rst fit:

arch the list of available space and allocate the first block that is big enough.

st fit:

arch the list of available memory space and allocate the smallest block that is big be enough.

orst fit:

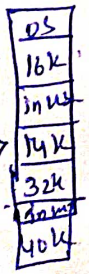
free storage list
5K
14K
16K
Req for 13K

arch the entire list of available memory and allocate the largest block.

rtition de allocation algorithm:

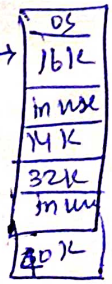
Best fit

Free area
5K
14K
16K
32K
40K
Req 13K



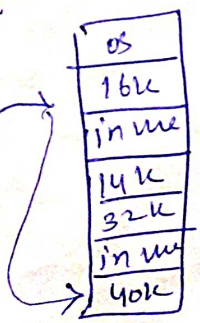
First fit

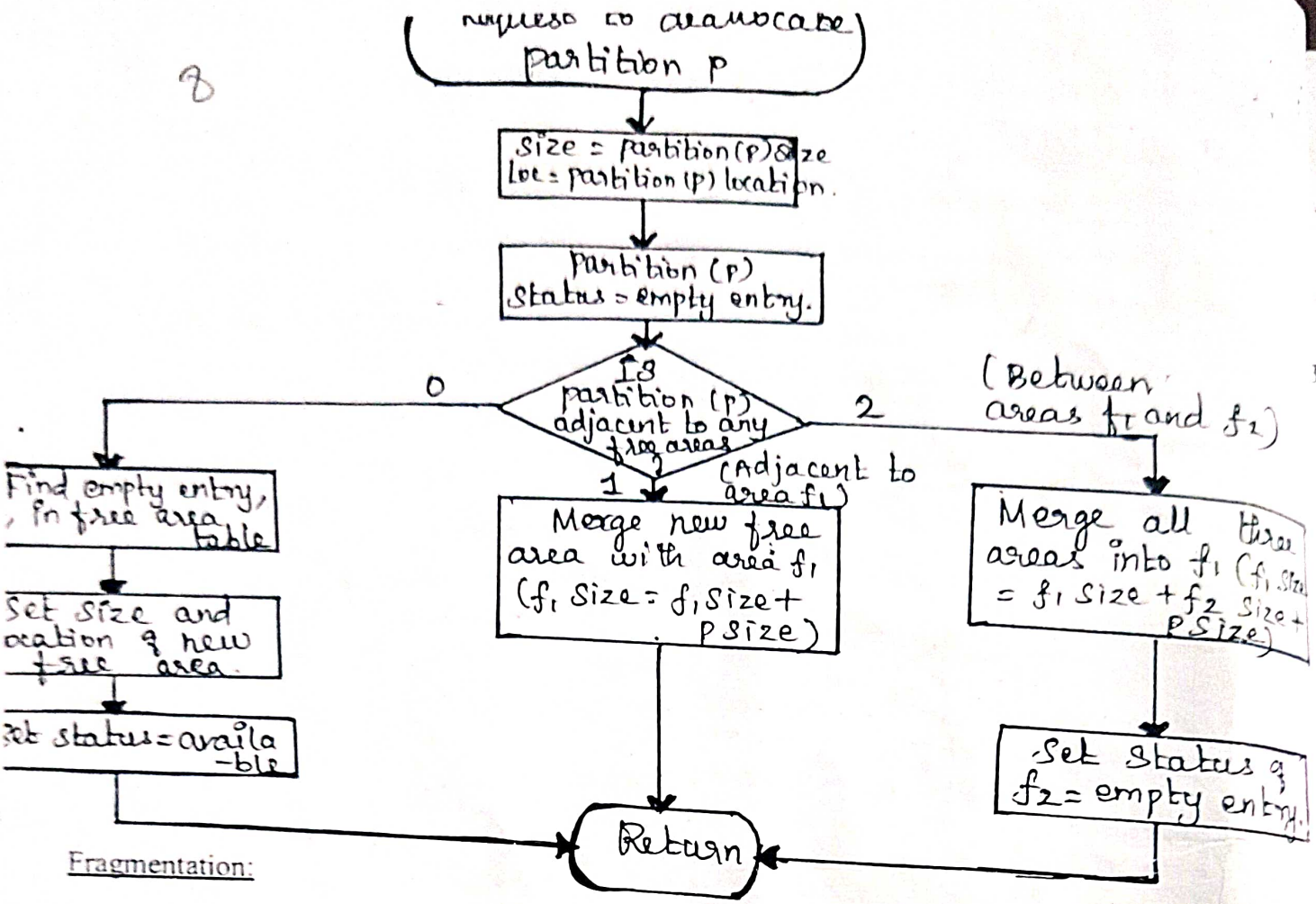
Req 13K



Worst fit

Req 13K





Fragmentation:

- Development of large number of separate free memory areas.

External fragmentation:

- Total memory space exist to satisfy a request, but it is not contiguous.

Internal fragmentation:

- Allocated memory may be slightly large than the requested memory..

Multiple partitioned algorithm:

- A single job may be allocated in more than one partition is called multiple partitioned algorithm. Sometimes it decreases the fragmentation.
- Eg: Assume a job requires 100KB of memory. The job can be allocated either 100KB portion or five 20KB partitions or two 40KB partitions and so on is called multiple partitioned algorithms.

Advantages:

- It allows multiprogramming.
- It increases efficient utilization of processors and i/o devices.
- It does not require special costly h/w.
- The algorithm used is simple and easy to implement.

Disadvantages:

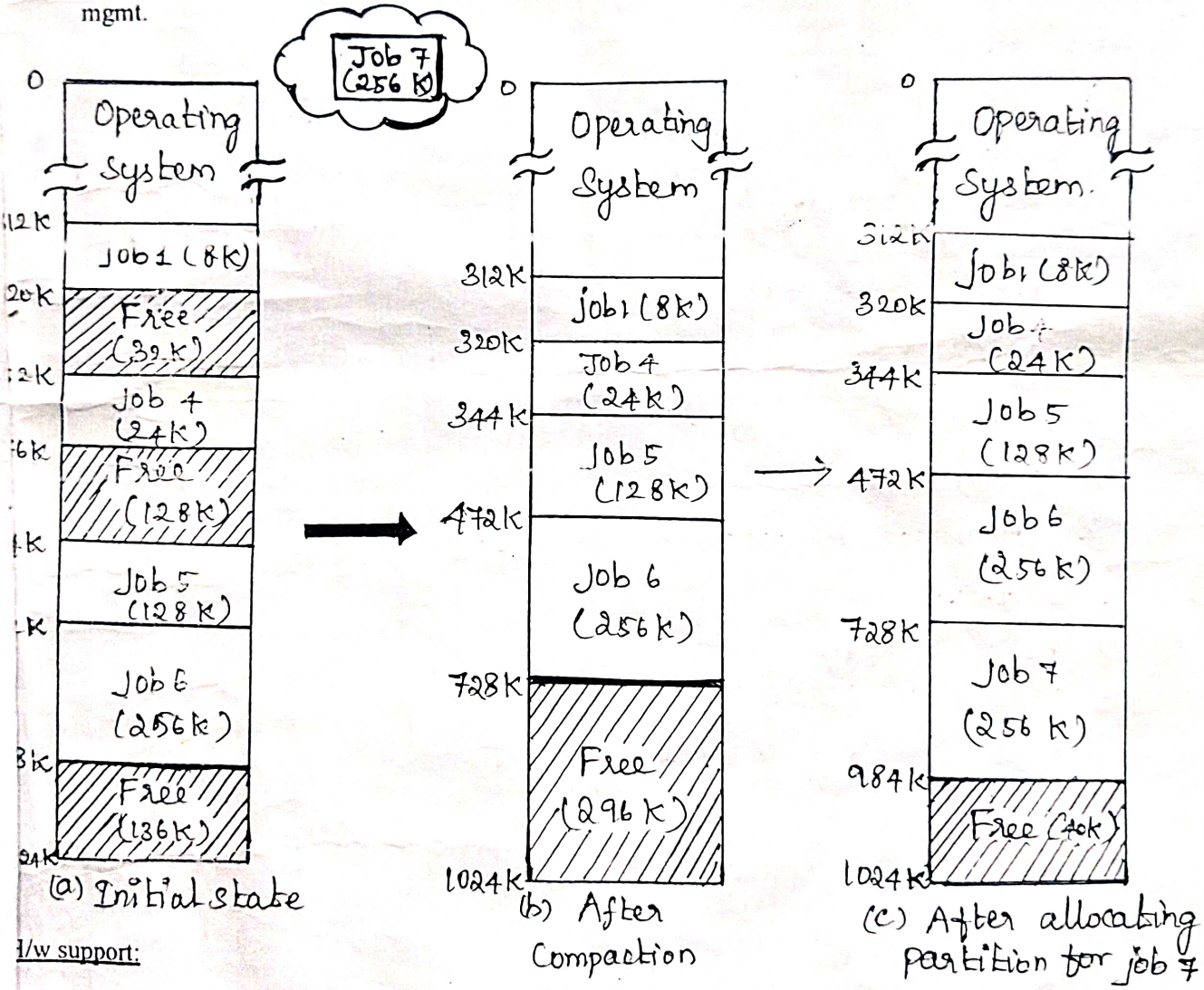
- Fragmentation can be an important problem.
- Even if may not be large enough for a partition
- Example: suppose we have 512k memory and a job size is 212 bytes are wasted. Since they are too small for any of the other jobs.
- It requires extra memory compare to single contiguous allocation.

3. **Re locatable partitioned memory mgmt:**

- This technique provides solution to the fragmentation problem.

Compaction or re compaction:

- Periodically combine all free areas into one contiguous area is called compaction.
- The process of adjusting job partition addresses is called re located partitioned memory mgmt.



H/w support:

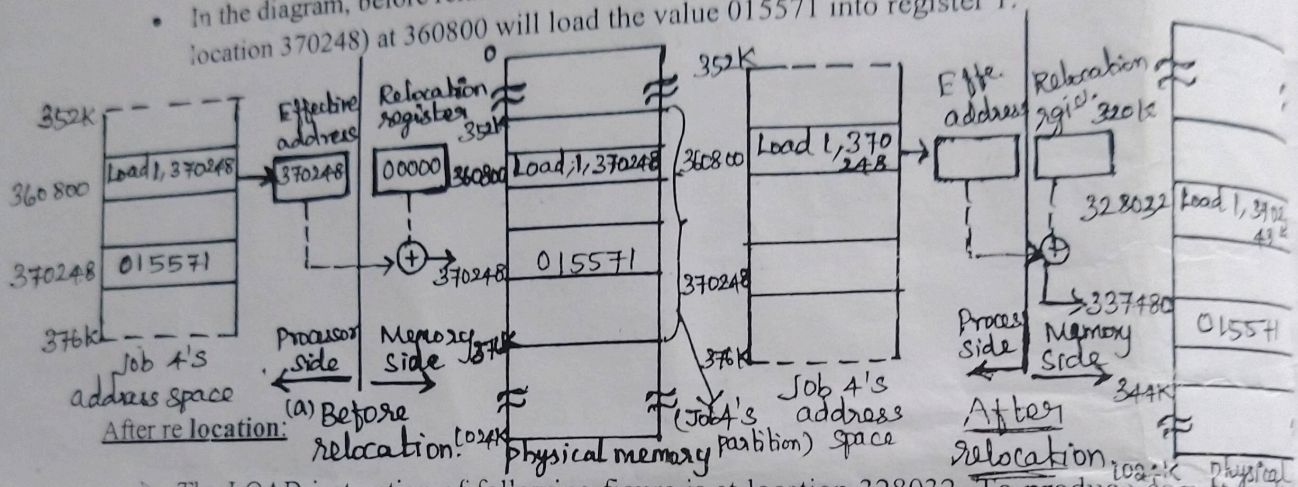
- The common approach to the re location problem is dynamic relation approach. In this approach two specialized privileged register are used.

1) Base relation register.
2) Bound register.

- On every memory reference, the content of the base relation register is automatically added to the effective address.

Effective address:

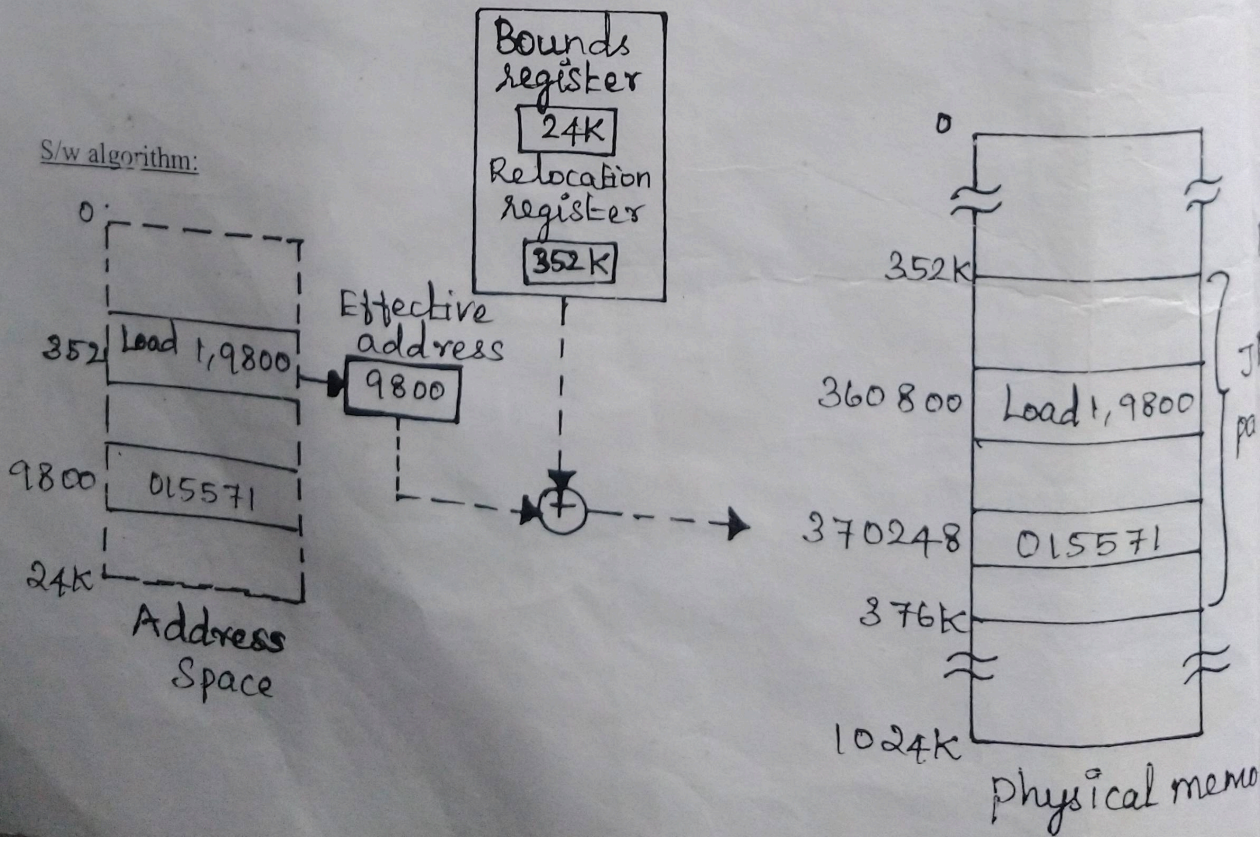
- The effective address is the final reference address (memory) computed by processor..
- In the diagram, before relation the instruction LOAD 1,370248 (LOAD register-1 from location 370248) at 360800 will load the value 015571 into register 1.



After re location:

- The LOAD instruction of following figure is at location 328032. To produce correct results The OS must set the re location register to -32768.
- When the instruction LOAD is encountered,-32768 is automatically added to the effective address 370248. To determine the actual physical memory location to be accessed.
- In this case 370248-32768=337480.thus the value 015571 is loaded into register 1.
- Re location adjustment is done automatically as each instruction is executed is called dynamic re location.

S/w algorithm:

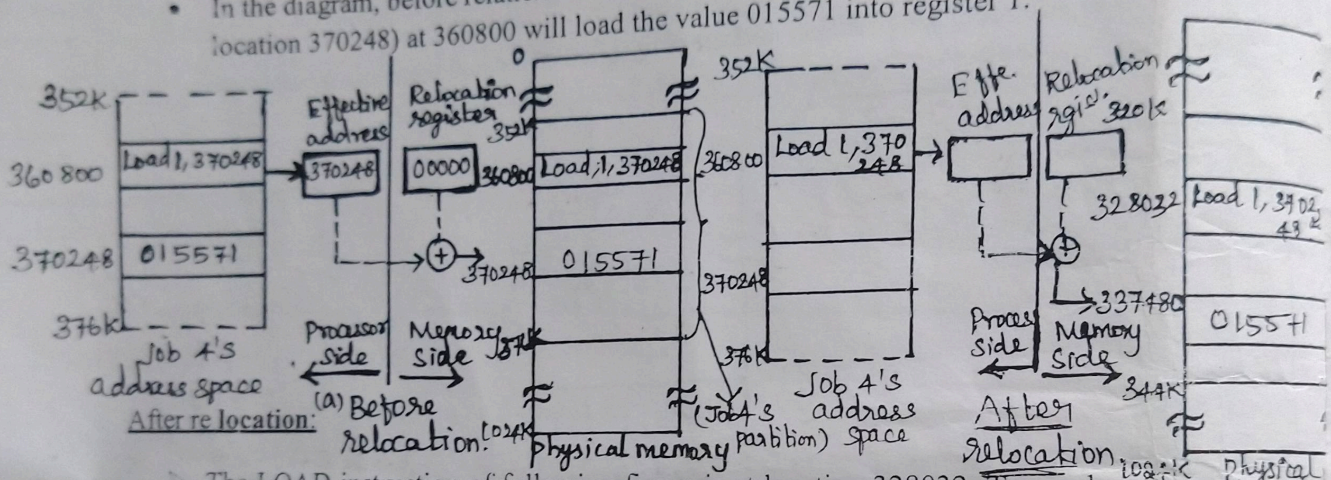


1) Base relation register.
2) Bound register.

- On every memory reference, the content of the base relation register is automatically added to the effective address.

Effective address:

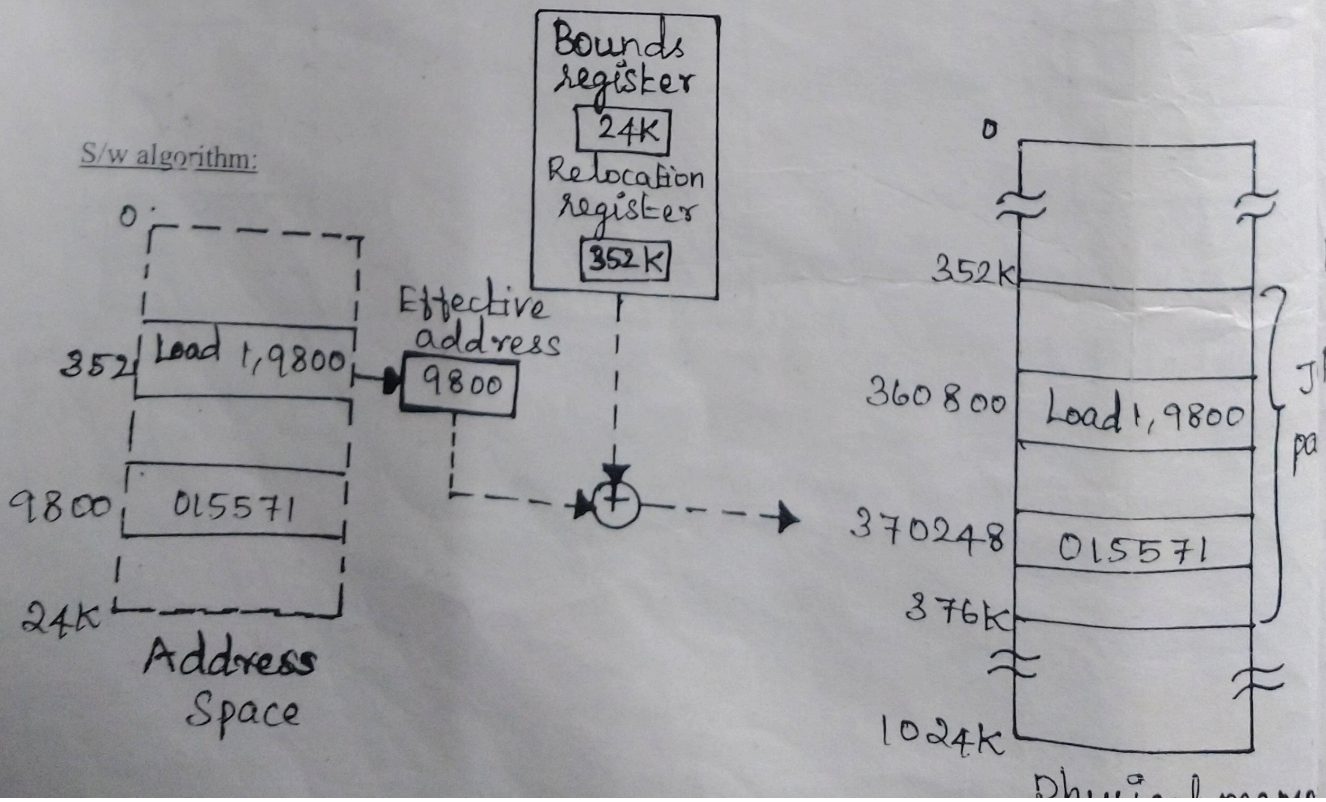
- The effective address is the final reference address (memory) computed by processor..
- In the diagram, before relation the instruction LOAD 1,370248 (LOAD register-1 from location 370248) at 360800 will load the value 015571 into register 1.



After re location:

- The LOAD instruction of following figure is at location 328032. To produce correct results The OS must set the re location register to -32768.
- When the instruction LOAD is encountered,-32768 is automatically added to the effective address 370248. To determine the actual physical memory location to be accessed.
- In this case 370248-32768=337480.thus the value 015571 is loaded into register 1.
- Re location adjustment is done automatically as each instruction is executed is called dynamic re location.

S/w algorithm:

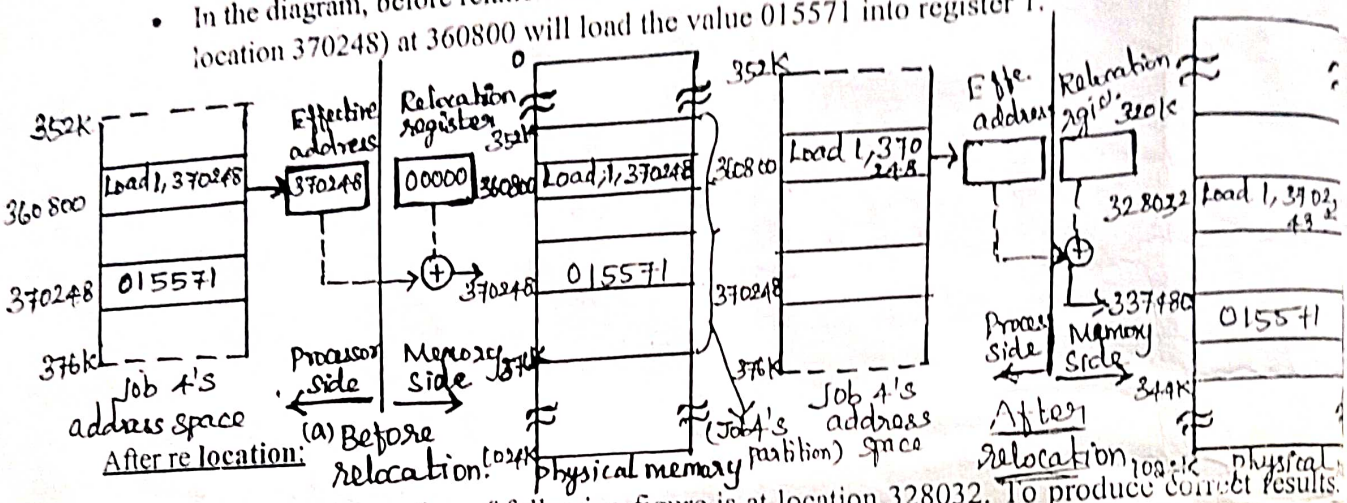


1) Base relation register.
2) Bound register.

- On every memory reference, the content of the base relation register is automatically added to the effective address.

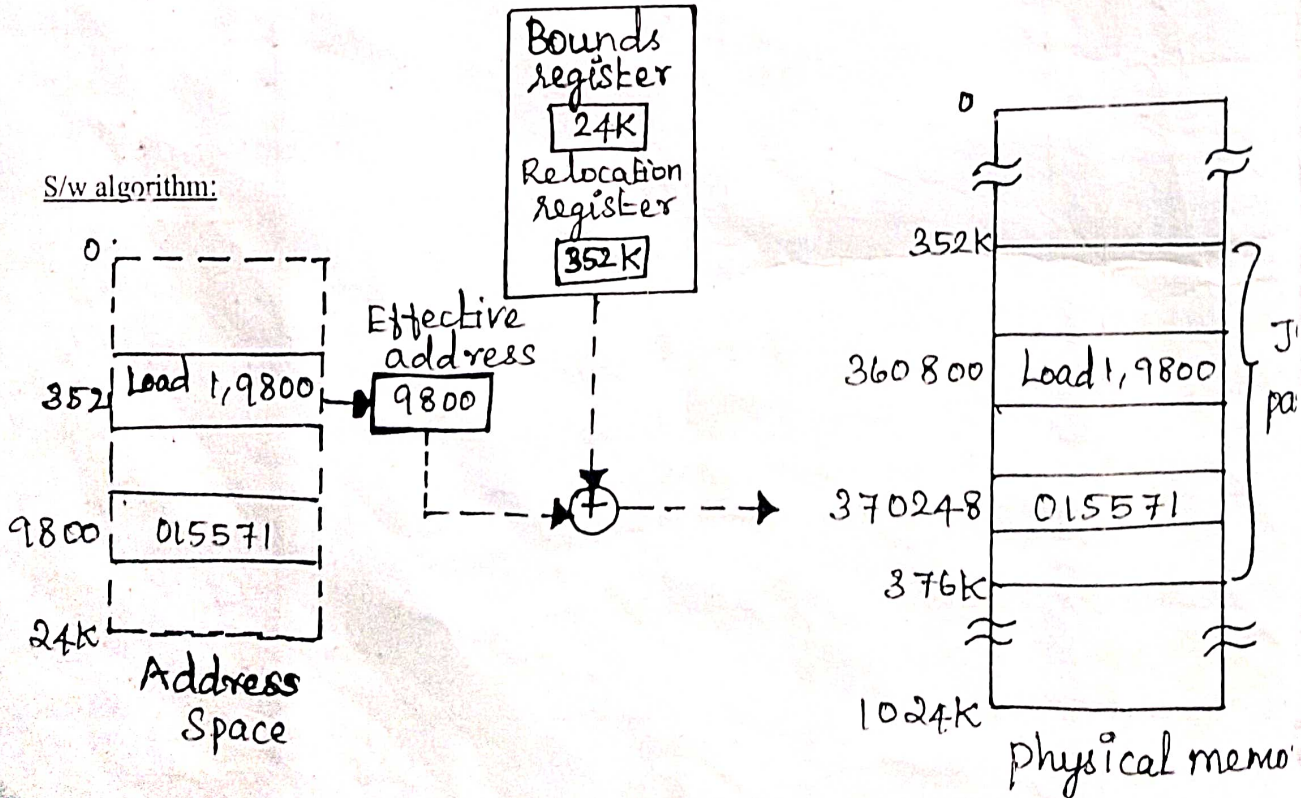
Effective address:

- The effective address is the final reference address (memory) computed by processor..
- In the diagram, before relation the instruction LOAD 1,5 /0248 (LOAD register-1 from location 370248) at 360800 will load the value 015571 into register 1.

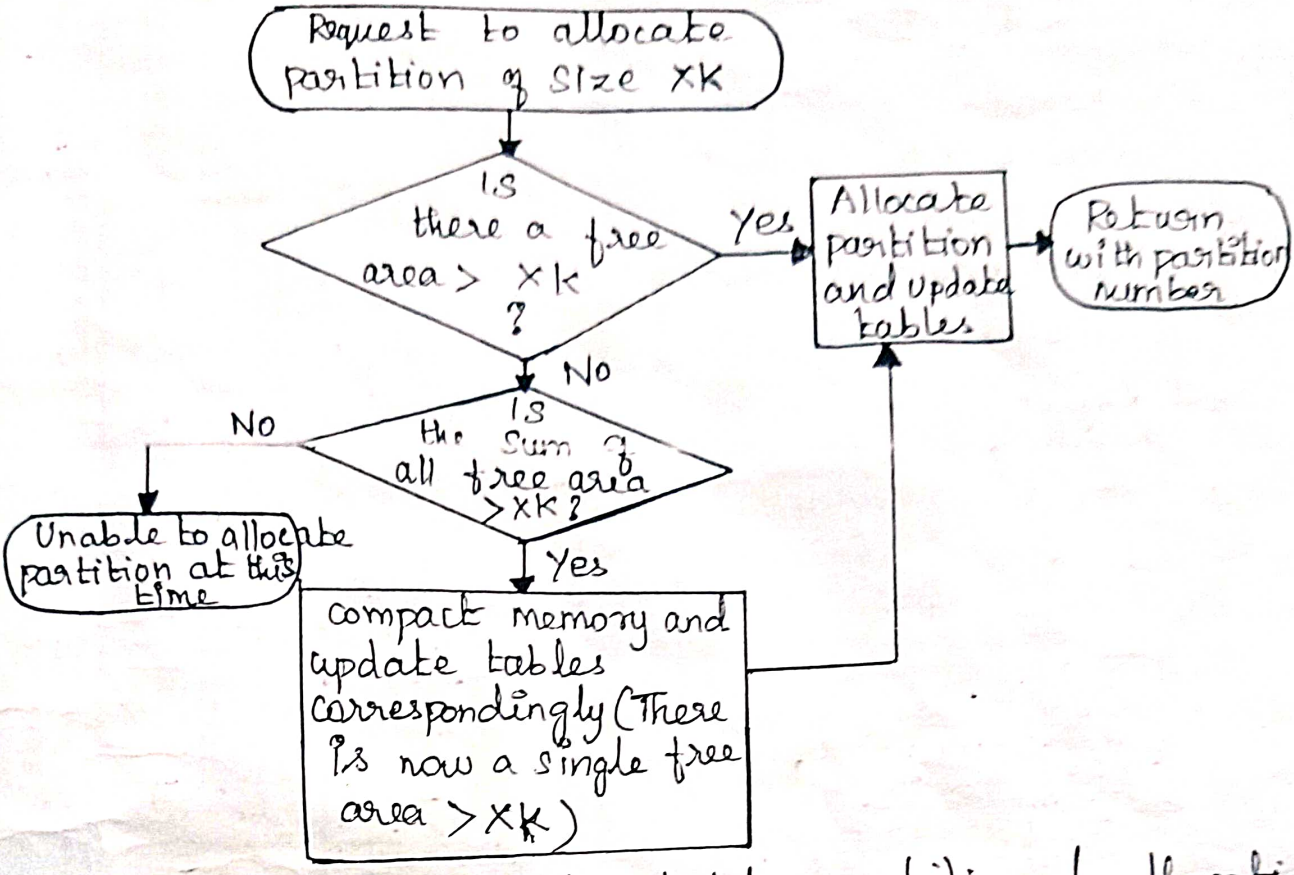


- The LOAD instruction of following figure is at location 328032. To produce correct results. The OS must set the re location register to -32768.
- When the instruction LOAD is encountered,-32768 is automatically added to the effective address 370248. To determine the actual physical memory location to be accessed.
- In this case 370248-32768=337480.thus the value 015571 is loaded into register 1.
- Re location adjustment is done automatically as each instruction is executed is called dynamic re location.

S/w algorithm:

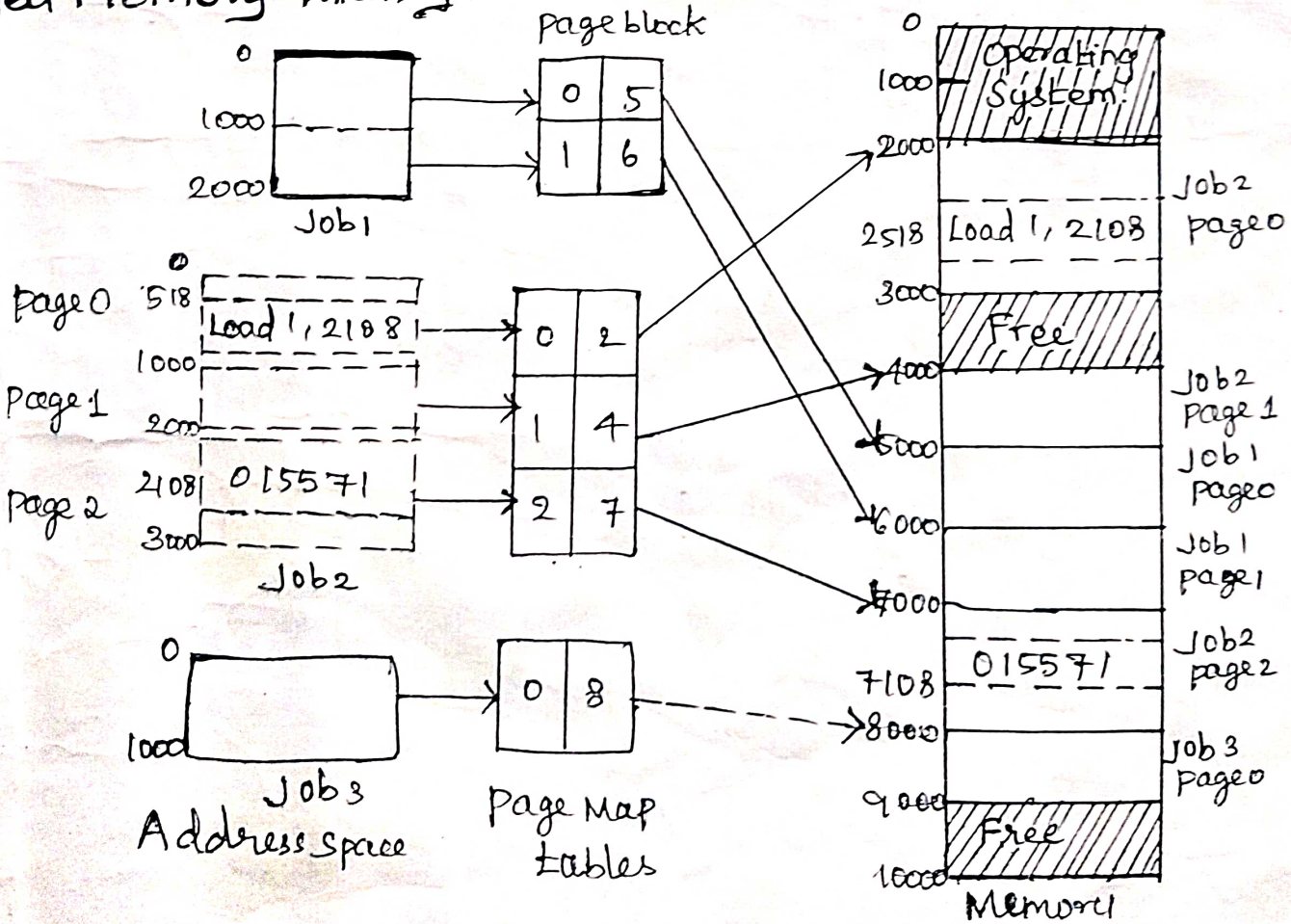


Relocatable partitioned allocation:

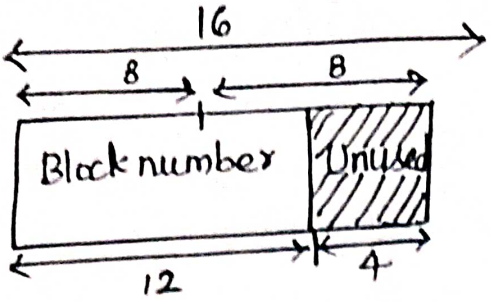


Overview of Relocatable partitioned allocation

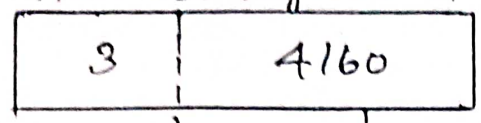
ged Memory Management:



12

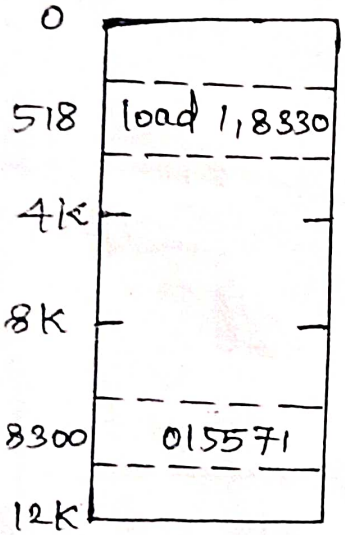


Page Map Table Address Register (PMTAR)

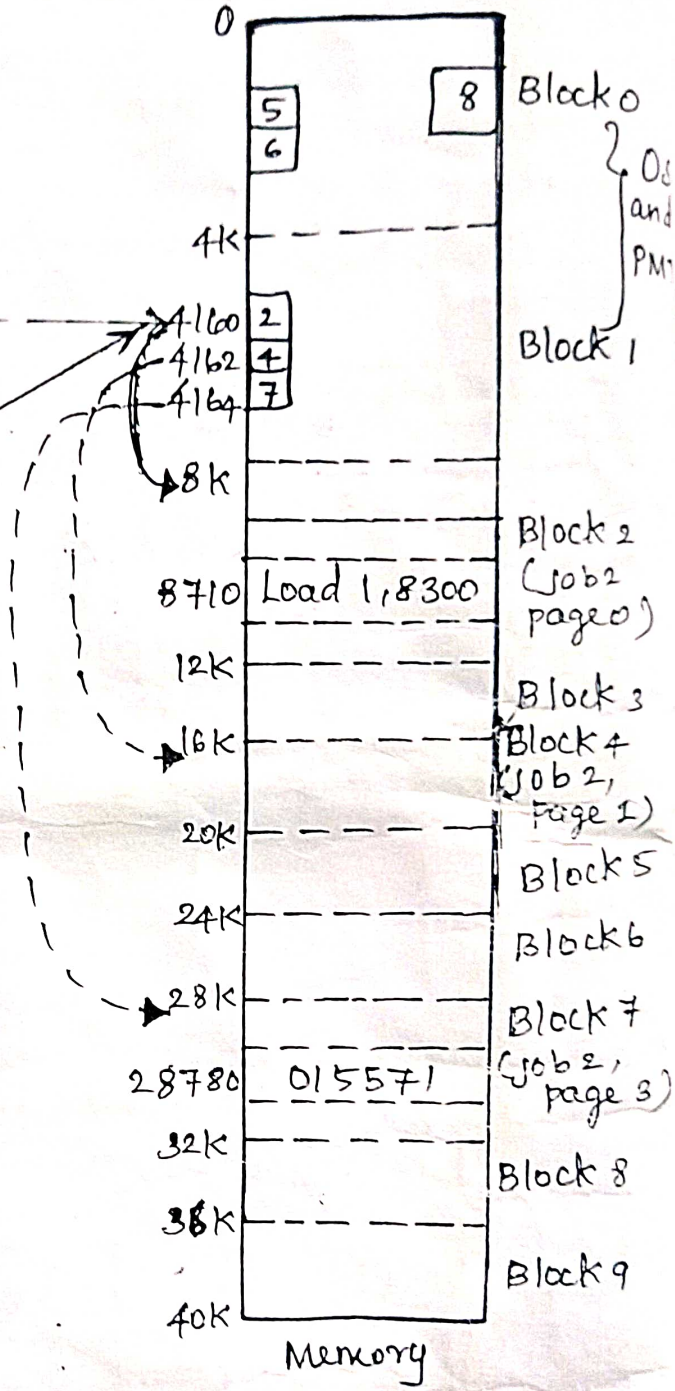


Length.

Address.



Job 2's Address space



Relationship between the pages, the blocks, the PMTAR, and the PMTs.

Advantages:

- Eliminates fragmentation problem.
- High utilization of memory and processor.
- Increase multiprogramming.

Disadvantages:

- Relocation h/w increases the cost of the computer.
- Compaction requires extra time.
- Some memory will still be unused because even though it is compacted, the amount of free area may be less than the needed job size.
- A jobs partition size is limited to the size of physical memory.

4. **Paged memory mgmt:**

- In paged memory mgmt each job's address space is divided into equal pieces called PAGES.
- Physical memory is divided into pieces of the same size called BLOCKS.
- Then by provide a suitable h/w mapping facility any page can be placed into any block.
- To perform the mapping from the job address space to physical memory using separate register for each page. These register are called PAGE MAP TABLES or PAGE MAP.
- This paged memory mgmt solves the fragmentation problem without physically moving partitions.

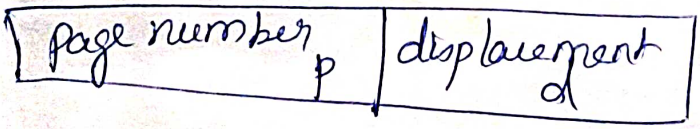
Jobs address space:

- If there were a job3 required 2000 bytes. we can assign job3's two pages to the available blocks, such as job3's page 0 is assigned to block 1 and job3's page 1 is assigned to block 1.
- H/w support in paging is illustrated in following figure:

A virtual address in paging system is an Ordered pair (p,d)

p → page number (Referred item resides)

d → displacement (at page p where referred item is located)



virtual address V=(p,d).

PMTAR:(Page Map Table Address Register)

- PMTAR indicate where the PMT is stored in physical memory.
- When the processor is switched to a new job only the PMTR has to be changed.t indicated the location of the new job's PMT.

HPMT:(Hybrid Paged Map Tabled)

In this scheme, combining the high speed mapping register and the PMT's to overcome this speed problem..

S /w algorithm:

- There are 3 basic table types that must be managed by the OS software.

Four function:

- Keeping track of the status of two tables.
  Paged map tables.
  Memory block tables
- Determine who gets memory.
- Allocate-each pages to physical memory.
- De allocates- when the job is finished block must be returned to free status.

H/w support:

- This scheme requires extra hardware. So it increases cost of computer. To avoid this high speed map register are used.

High Speed Paged Map Register:

- If the pages are increased cost of buying registers is also increased .To avoiding this we should go for high speed resister scheme?

PMT: (Page Map Table)

- It store in physical memory.
- Pmt determines the physical memory address from effective address.
- Pmt indicates page number of job's address space and block number of physical memory address.

a) Job table
b) Memory block table
c) Page map table

Job table:

- It indicates the job number, size, location of PMT and status information about the jobs.

Memory block table:

- It indicates the status of each memory block either allocated or available.
- These tables represented as follows,

Job Table

Job number (JT)	Size	Location of PMT	Status
1	8k	300	Allocated
2	4k	400	allocated
3			Empty entry

Block no.

3
7

Job 1's PMT

page no 0(400)

Block no

6

job2's PMT

Page Map Table (PMTs)

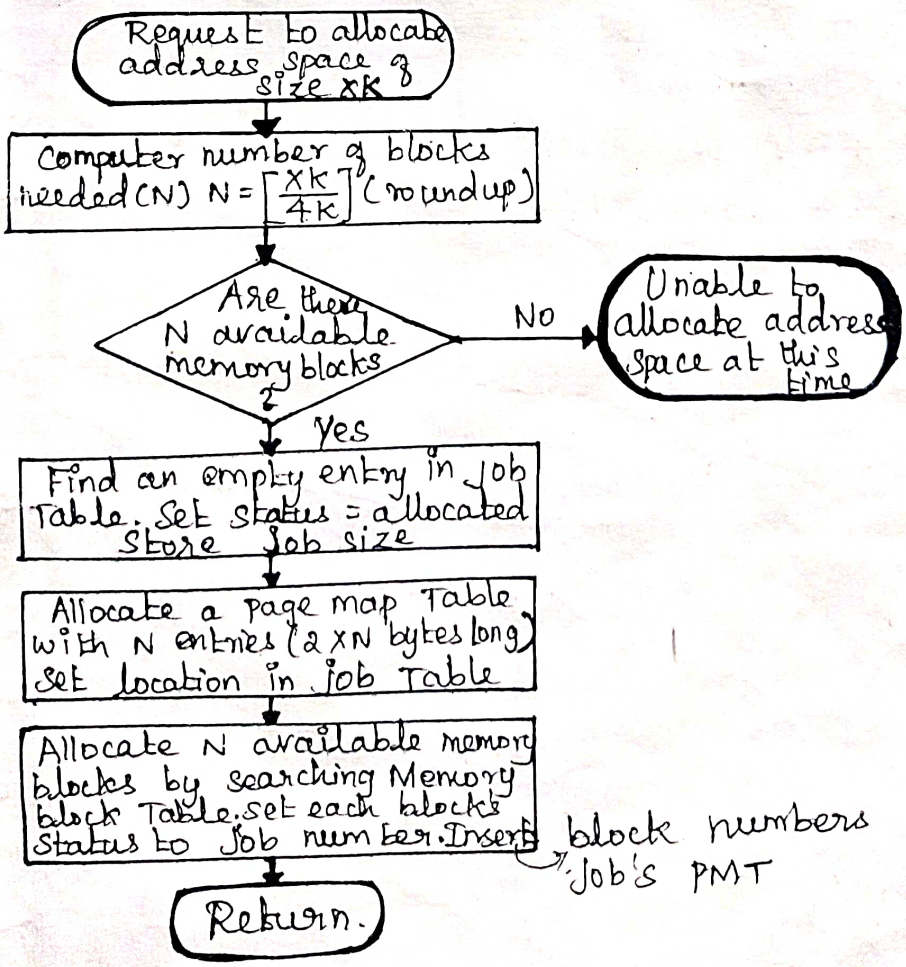
Page Map table

Block no	Status
0	Os
1	Os
2	Available
3	Job1
4	Available
5	available
6	Job2
7	Job1

Memory block table (MBT)

Page No	Block no
0 (300)	3
	7

Paged memory allocation algorithm or address space allocation:



Advantages:

- There is no fragmentation problem.
- Allows higher degree of multiprogramming.
- Higher utilization of processor memory and i/o devices.
- There is no need compaction.

Disadvantages:

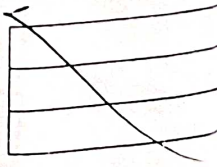
- It requires extra page address mapping hardware.
- It increases the cost of computer.
- Memory must be used to store the various tables such as PMT, PMART,JT etc.
- Processor time must be increased to maintain and update these tables.
- Internal fragmentation or page break does occur, some memory is unused.

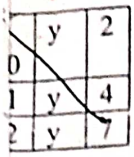
Note:

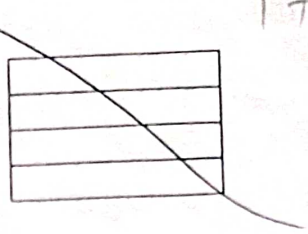
- ✓ In the previous schemes,
  A job could not be run until there was sufficient memory to load its entire address space.
- ✓ The problem could be resolved by using extremely large main memories with an illusion called *virtual memory.*
- ✓ There are two virtual memory techniques:
  i)Demand paged memory management
  ii)Segmented memory management

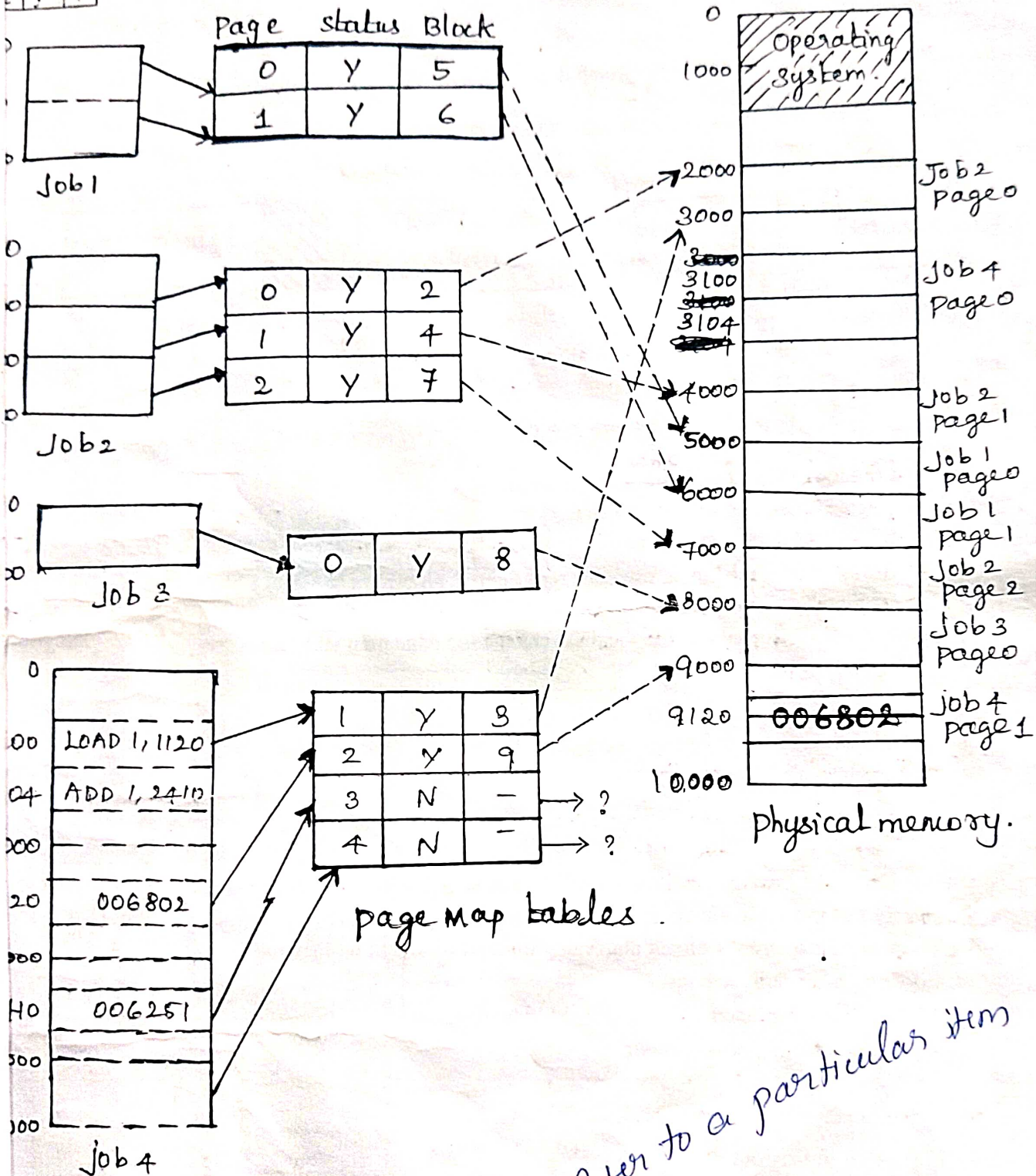
5. Demand paged memory management:

- It is one of the virtual memory management techniques.
- It is similar to paged memory management with swapping.
- Demand paged memory management can logically attain utilization greater than 100% of main memory.
- That is, the sum of all the address spaces of the job being multiprogrammed may exceed the size of the physical memory.
- In the following figure, the address mapping h/w encounters a page map table entry with status=N, it generate a page interrupt page fault.
- The OS must process the page interrupt by loading the required (demand) page and adjusting the page map table entry as y.
- This situation is known as demand paged memory management.









→ Refer to a particular item

Block Number b	Displacement d

virtual address v = (b, d)

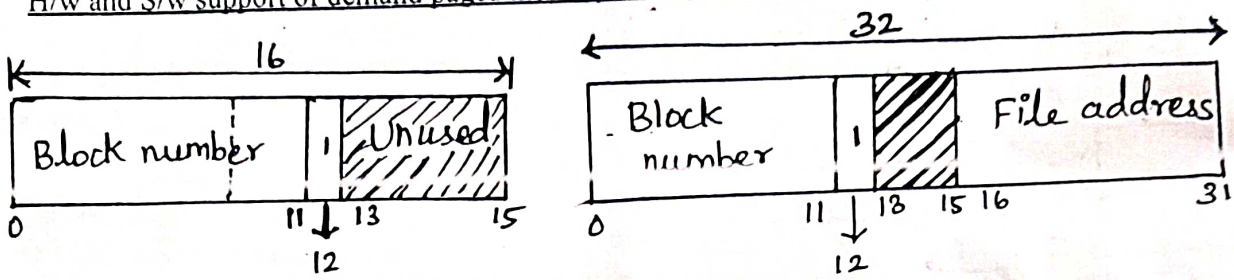
→ Start of the block.

18

0	Y	3
1	n	

Demand paged memory mapping

H/w and S/w support of demand paged memory management:



Page replacement takes the following approach:

- Find the location of the diamond page on the disk
- Find a free block
  a) If there is free block use it
  b) If there is no free block, use page replacement algorithm to select a block (that is not used currently).
  c) Write the selected page to the disk, change the page map table accordingly
- Read the demanded page into free block, change the page map tables.
- Restart the user process.

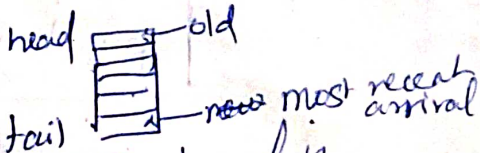
Page replacement algorithm:

- The OS select the page replacement algorithm with lowest page fault rate.
- Number of available blocks increases the number of page faults decreases.
- There are 3 page replacement algorithm are used by OS to replace pages.
  1) FIFO page replacement
  2) LRU page replacement
  3) Tuple coupling replacement

1.FIFO page replacement algorithm:

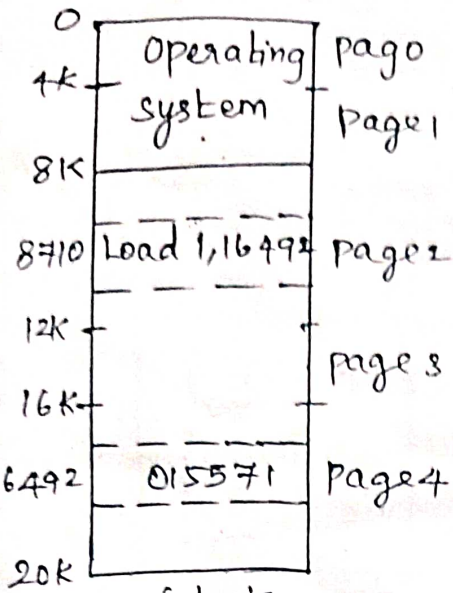
- FIFO-first in first out
- In this algorithm pages are replaced first in first out manner (removes the page that has been in memory for long time)
- It eliminates the possibility of loading a page and immediately removing it.
- P is the number of page reference
- M represent memory size

Failure reference function f=f/|p|

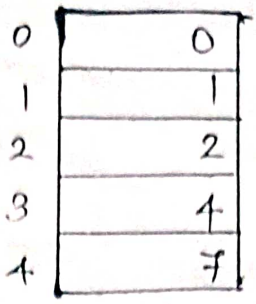


On a page fault, the page at the head is removed, & the new page added to the tail of list.

Overlapping address spaces. (Paged Memory Management)

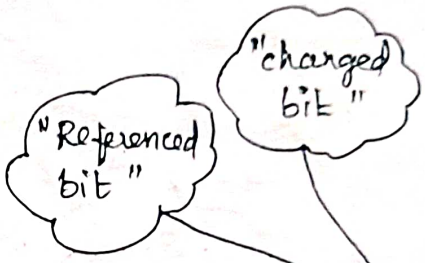


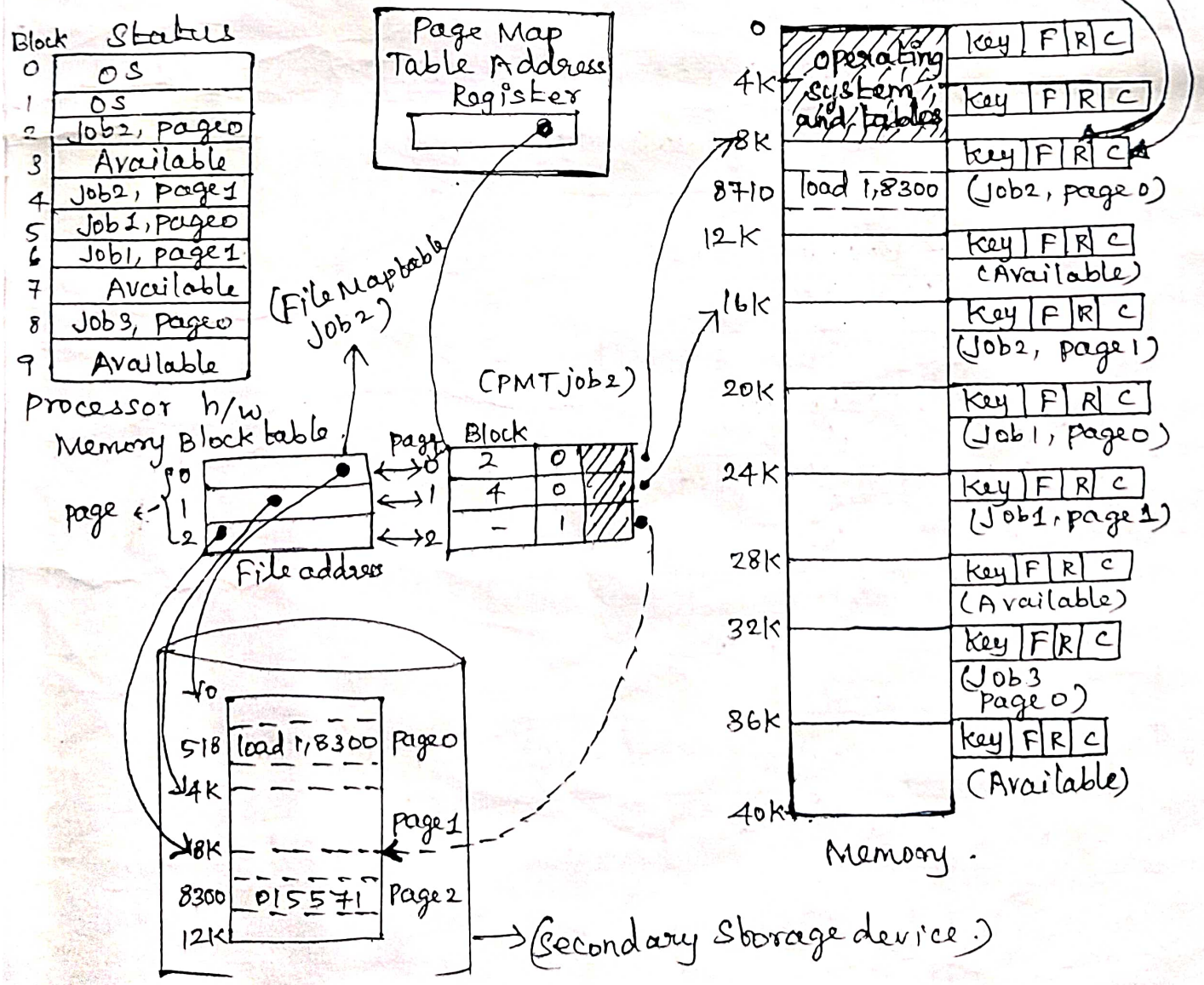
Job 2's address space.

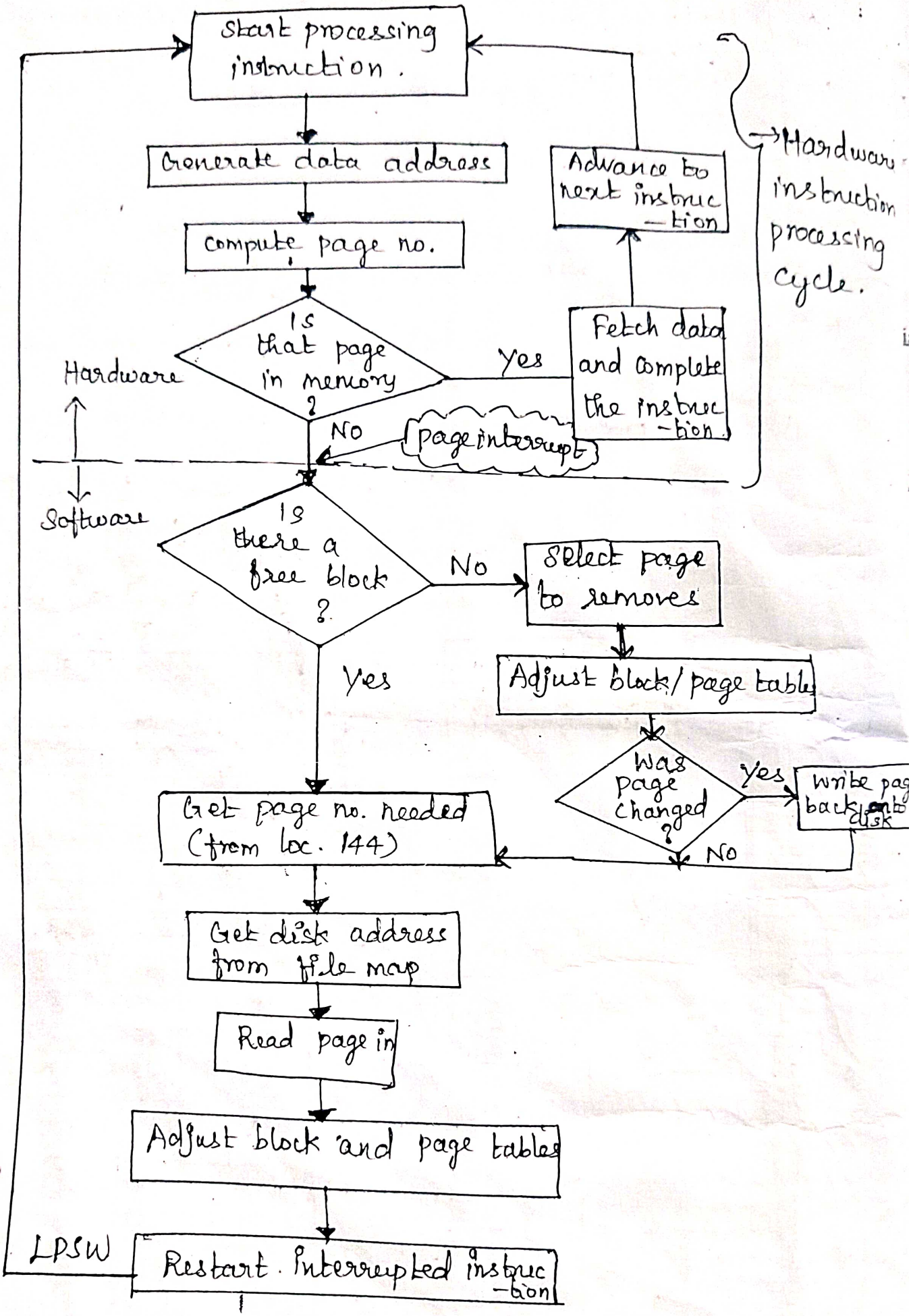


page Map table.

Demand paged Memory Management:







Interaction between h/w and s/w.

Example: m=3 represent memory size

P indicate the page trace/order

4	3	2	1	4	3	5	4	3	2	1	5

4*	3*	2*	1*	4*	3*	5*	5	5	2*	1*	1
	4	3	2	1	4	3	3	3	5	2	2
		④	③	②	①	4	4	④	③	5	5

Failure frequency function f= F/|p| = 9/|12| =75%

Success frequency function s= s/|p| = 3/|12| = 25%

Eg: 3 frames

Reference string is

1,2,3,4,1,2,5,1,2,3,4,5

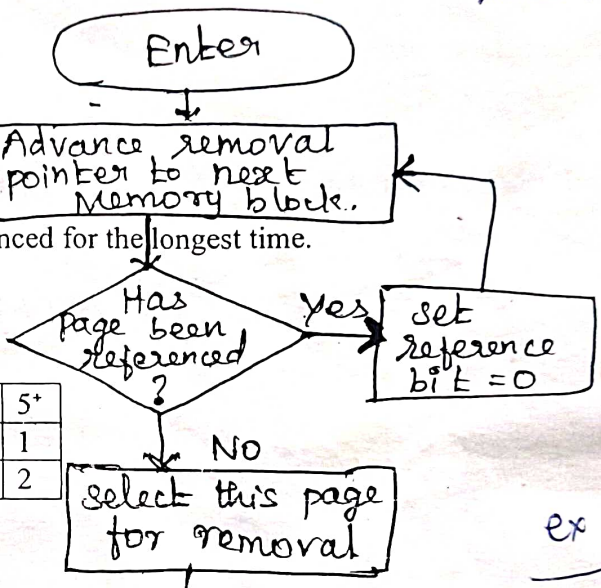
12 pages in string

Total 9 page fault

Hit = 3.

Disadvantage:

- If a page becomes the oldest and will be removed, even though it will be needed again immediately.
- Some strong side effect can occur (FIFO anamol).



LRU page replacement algorithm:

- LRU-Least Recently Used.
- LRU select for removal the page that has not been referenced for the longest time.

Example:

4	3	2	1	4	3	5	4	3	2	1	5

4*	3*	2*	1*	4*	3*	5*	4	3	2*	1*	5*
	4	3	2	1	4	3	5	4	3	2	1
		④	③	②	①	4	3	⑤	④	③	2

+	+	+	+	+	+	+			+	+	+

F=10

F=10/|12|=83%

ex for LRU

LRU approximation algorithm.

TUPLE coupling replacement algorithm:

It is a simple approach

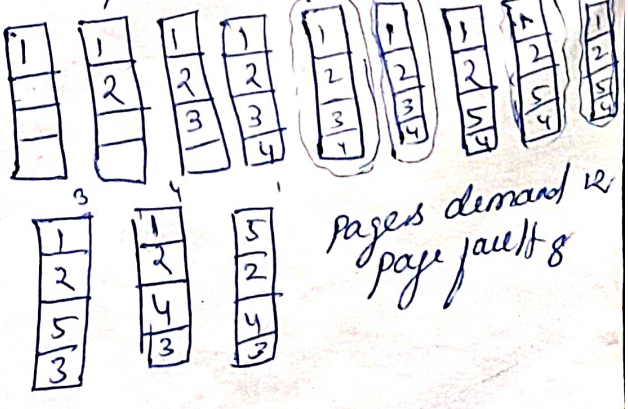
- Each original page p must be divided into two tuple such as p' and p".
- The removal ordering policies must be applied to both of the tuple so that page p'/p" is never removed unless the corresponding large page p would also have been removed from m.

1'	2'	1"	2"	3'	3"	2'	1'	1"	3'	3"

arrange p" then p'

1'*	2'*	1"*	2"*	3'*	3"*	2'	1'*	1"*	3'*	3"*
	1'	1'	2'	2"	3'	2"	2'	1'	1"	3'
		2'	1"	2'	2"	3"	2"	2'	1'	1"

Assume 4 frames & reference string as 1,2,3,4,1,2,5,1,2,3,4,5

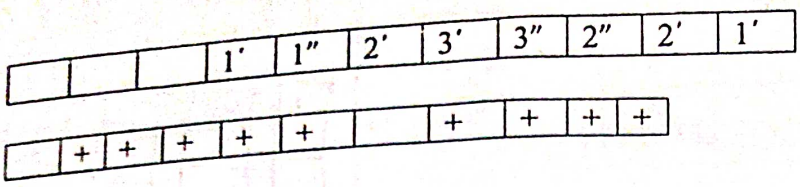


Pages demand 12

page fault 8

Implementation – LRU

- Needs substantial h/w support
- problem to determine an order
- 2 imple feasible : stack
  : counter



F=10
F=10/|11|=91%

Locality: - concept is that processes tend to reference storage in non uniform highly localized patterns

- ❖ Temporal locality
- ❖ Spatial locality

Temporal locality:

Once a location of a program is referenced, it is often referenced again very soon.
eg. loop, stack, variables used for counting & totalling

Spatial locality:

It refers to the probability that once a location is referenced, a nearby location will be referenced soon. eg: Array traversal

Advantages:

- ➢ Fragmentation is eliminated.
- ➢ Compaction is not necessary
- ➢ Large virtual memory-a job's address space is able to longer than the size of physical memory
- ➢ More efficient use of memory: The portions of a job address space that are not used due to error routines-that will no kept in physical memory.
- ➢ It allows unlimited multi-programming.

Disadvantages:

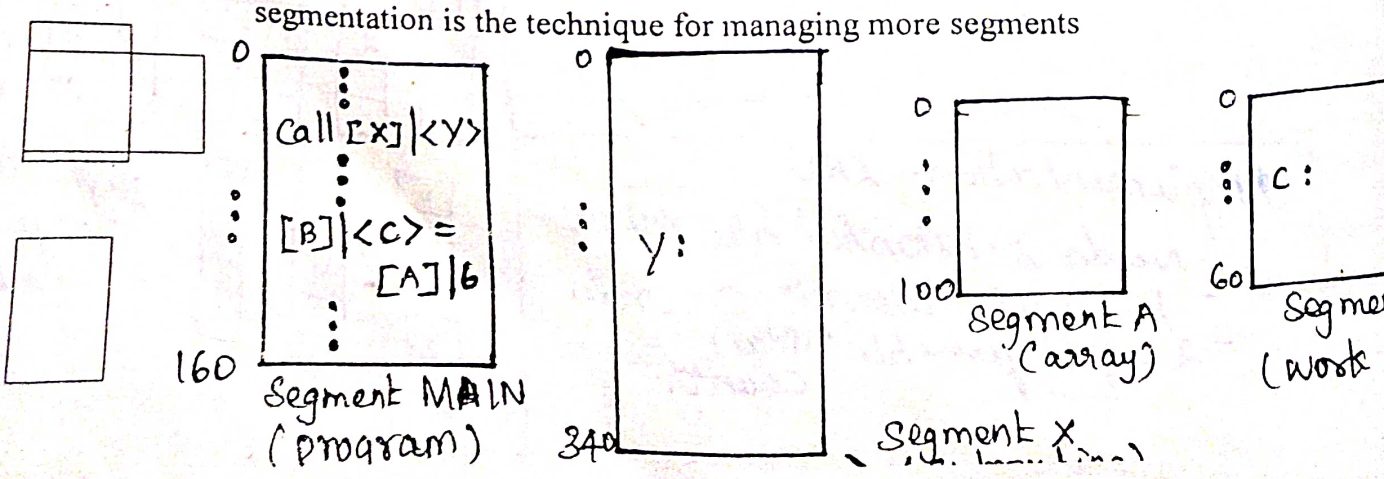
- ➢ It requires extra h/w.
- ➢ It need page breakage.
- ➢ Processor overhead.
- ➢ Memory space for table.

6. Segmented memory management:

Segment may indicate procedure, or array, or stack or collection of scalar vars

- ➢ A segment can be divided as a logical grouping of information (such as subroutine, array/da area).
- ➢ Thus each job's address space consist of a collection of segments as illustrated in the figure

segmentation is the technique for managing more segments

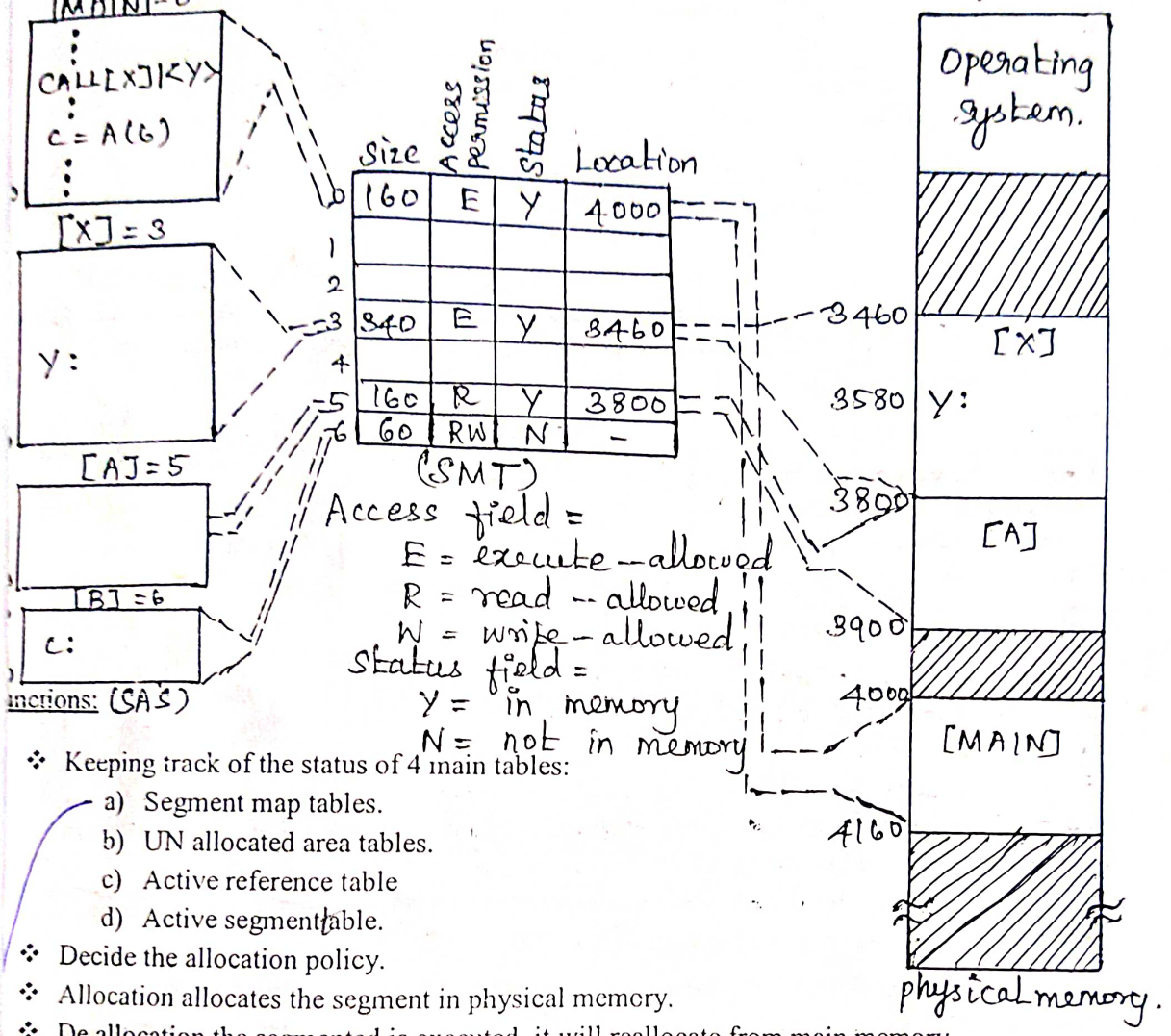


SAS : Segmented address space

Call[x]/<y>: transfer to entry point y within subroutine x

Call[a]: call the array a segment

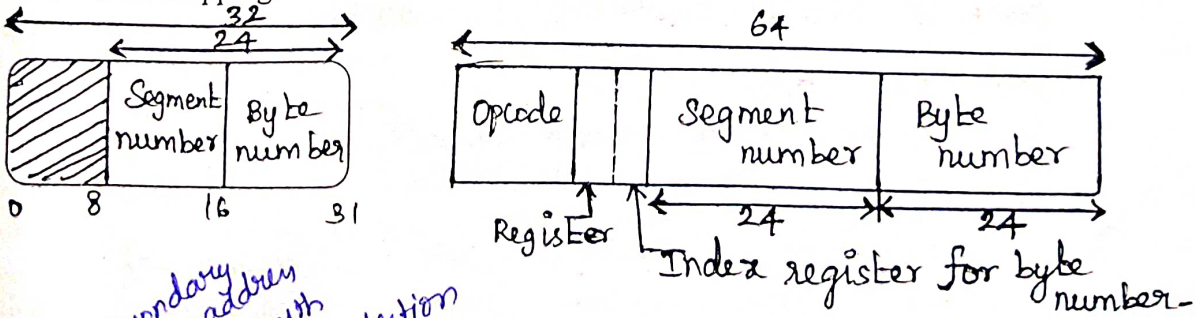
- SMT: (segment map table) that indicate the address of each segment in memory. → Append (A) this segment may have info added to it as it is end
- It contain segment size, access facility(executable(E),read(R),write(W),status(Y=segment in main memory)(n=segment not in main memory)and location of physical memory address.

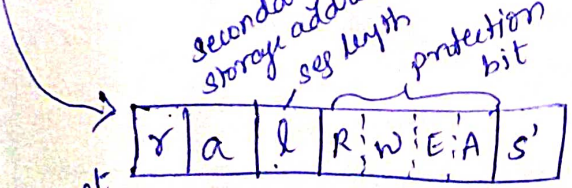


functions: (SAS)

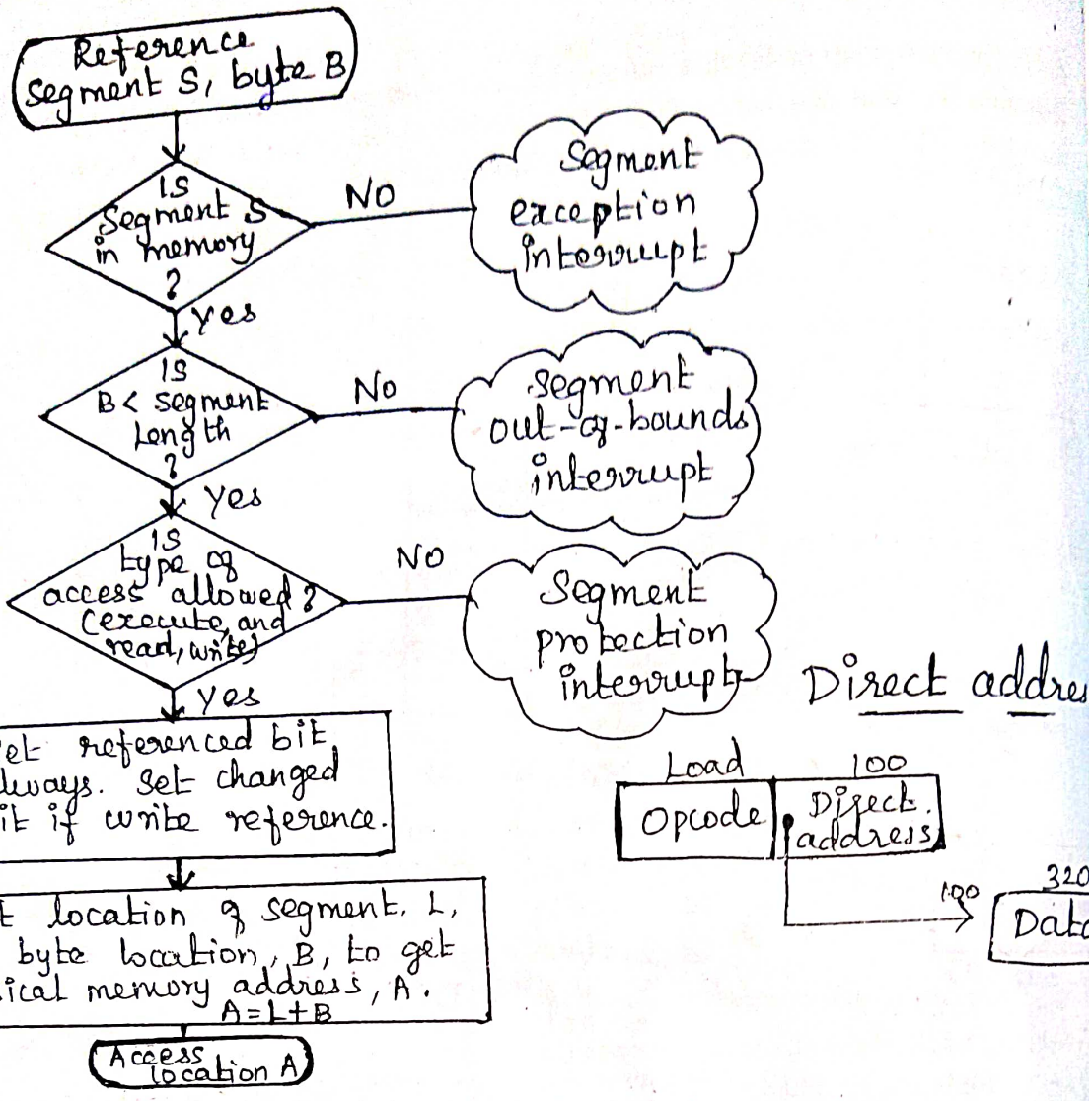
- ❖ Keeping track of the status of 4 main tables:
  - a) Segment map tables.
  - b) UN allocated area tables.
  - c) Active reference table
  - d) Active segment table.
- ❖ Decide the allocation policy.
- ❖ Allocation allocates the segment in physical memory.
- ❖ De allocation the segmented is executed, it will reallocate from main memory.

Segmented address mapping:



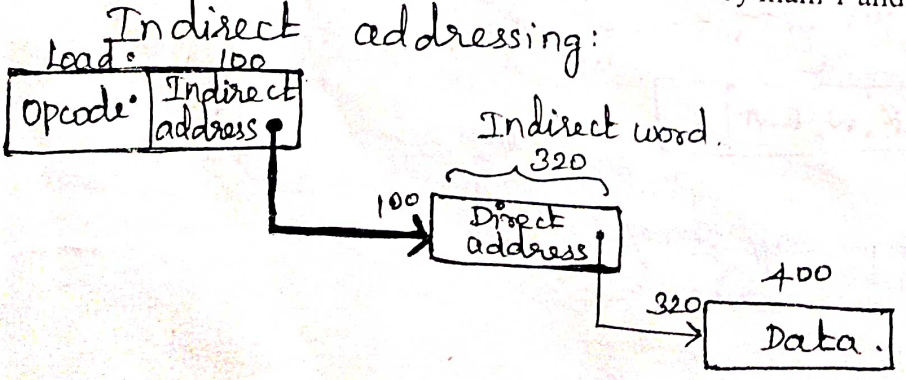


Formation of Segmented Address

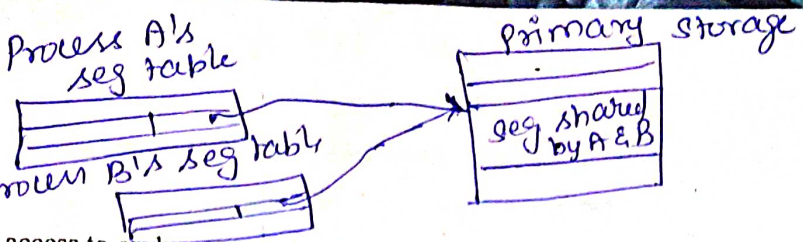


Advantages:

- Elimination of fragmentation: by moving segments in physical memory, fragmented memory space can be combined into a single free area
- Provide virtual memory: by keeping only the currently used segments in main memory, the job's total address space size may exceed the physical memory size.
- Allowing dynamically growing segments: if a segments size must be increase during execution, the out of range reference can be detected by out of bound interrupt occurs. The os must increase the length of the segment. Finally the segment table adjusted.
- Dynamic linking and loading: In the linking and loading environment the main prg is initially loaded, if it references any other procedures those segments are loaded and addressing links established at the time of reference. This is called de reference linking / dereferred linking. Thus by dereferring unnecessary linking is avoided.
- Shared segments: a segment may be used by different jobs, it is wasteful to have two separate copies exists main memory. Instead of copies the same segment in different place, sharing the segment by all the place, sharing the segment by all required job is called shared segments. In the example segment has been shared by main 1 and main2 procedures.



Sharing in segmentation



- Enforced control access: it access to each segment should be controlled by segment map table access control information entry((I . e R-read only ,w-write only, RW-read write only)

Disadvantages:

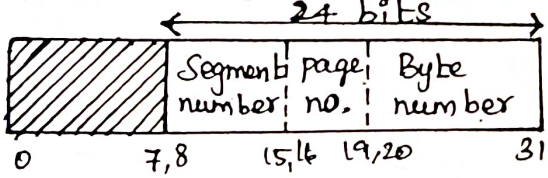
- Increase the h/w.
- Need additional memory for store tables.
- Compaction overhead occurs due to dynamic segment growth.
- There is difficult in managing variable size segments on secondary storage.
- Maximum size of the segment is limited by the size of main memory.

Segmented and demand paged memory management:

this scheme combine both the segmented and demand paging memory management mechanisms. ach segment is subdivided into number of pages.

S/w support:

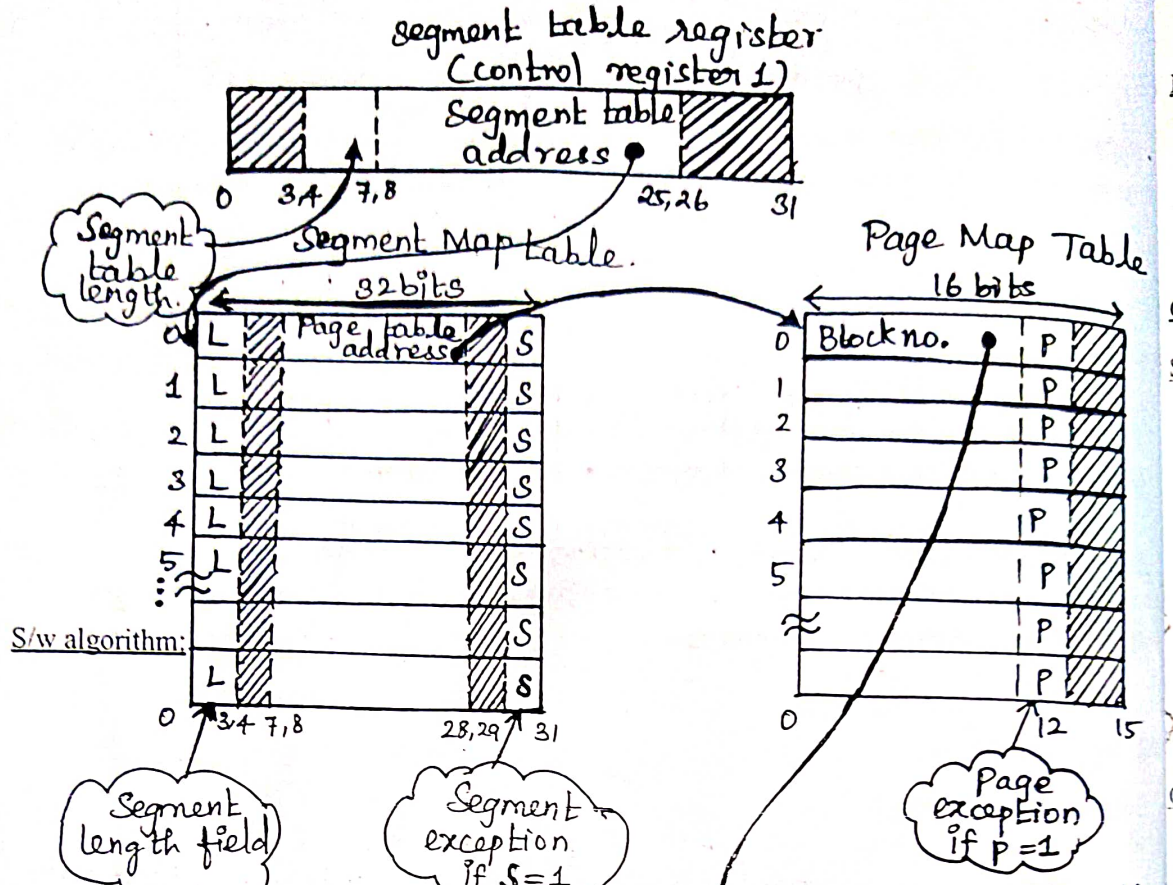
1) Segment table register(STR)
   - It is a 32 bit register.
   - It indicates the location and length of the current segment map table.
2) Segment map table(SMT)
   - It is a 32 bit register.
   - Each entries of this table indicate location and length of a page map table.
3) Page map table(PMT)
   - It is a 16 bit register.
   - Each entries of this table indicate the main memory block number for each page of that segment. This are depicted in following figure:

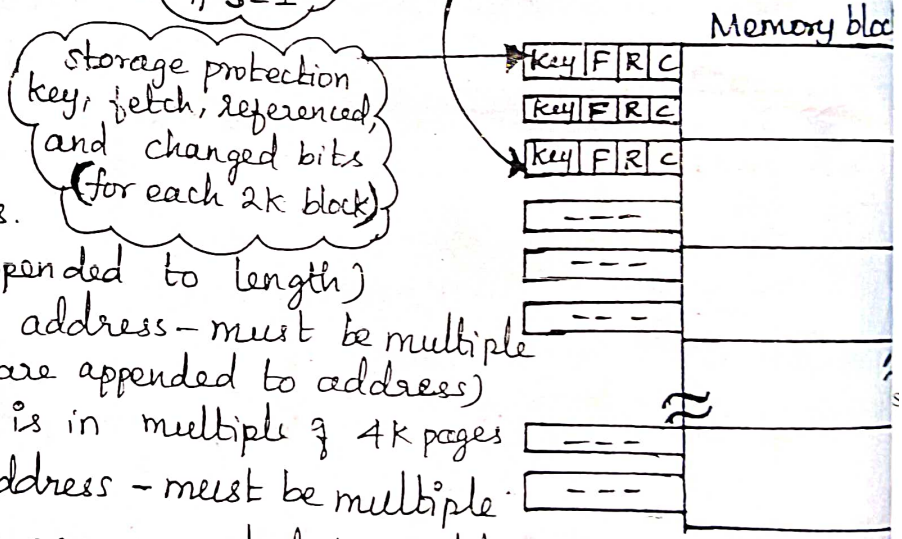


∴ The mapping from virtual address to physical address is accomplished by a series of tables.

∴ To decrease hardware costs as well as save bits in the various table entries certain restrictions are imposed.

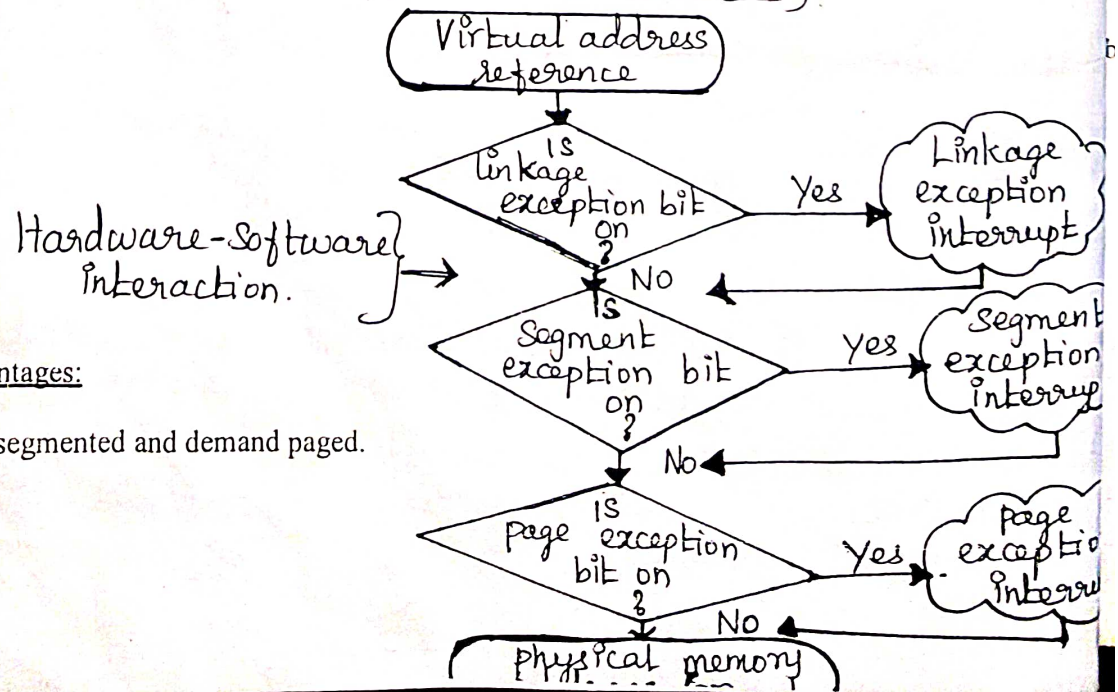
∴ The 24 bit 370 effective address is split into three parts in fig.





Notes:

1. Segment table length - must be multiple of 16 entries. (four zeros are appended to length)
2. Segment table address - must be multiple of 64 (six zeros are appended to address)
3. Segment length is in multiple of 4K pages
4. Page table address - must be multiple of 8 (three zeros are appended to address)



Advantages:

Both segmented and demand paged.

Disadvantages:

- Need page breakage.
- Extra memory needed for SMT and PMT.
- Increasing h/w cost.
- Complexity can be quite significant.

Other memory management schemes:

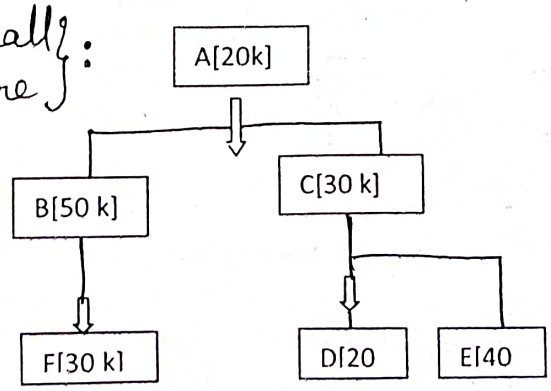
Swapping:

a) The early compatible time sharing system uses this swapping techniques with single contiguous allocation scheme, only one job was in main memory at a time. After running for a short period. The current job was swapped into secondary storage.
b) In the similar manner the partitions of a system can be swapped.
c) If a high priority job arrives for processing and insufficient main memory is available, one or more jobs removed from main memory and swapped onto secondary storage is called rollout.
d) When the high priority job has completed. The low priority job may be reloaded into main memory is called rolling.

Overlays:

a) Swaps only portions of the jobs are called overlay management.
b) The figure illustrates the relationship between the procedures in a job's. This diagram indicates that procedures A calls only procedure B and C, procedure B calls only procedure F, procedure C calls only D and E.
c) We notice that procedure B never calls C and C never calls B.
d) Thus procedures B and C do not have to be in memory at the time.

procedure call structure:

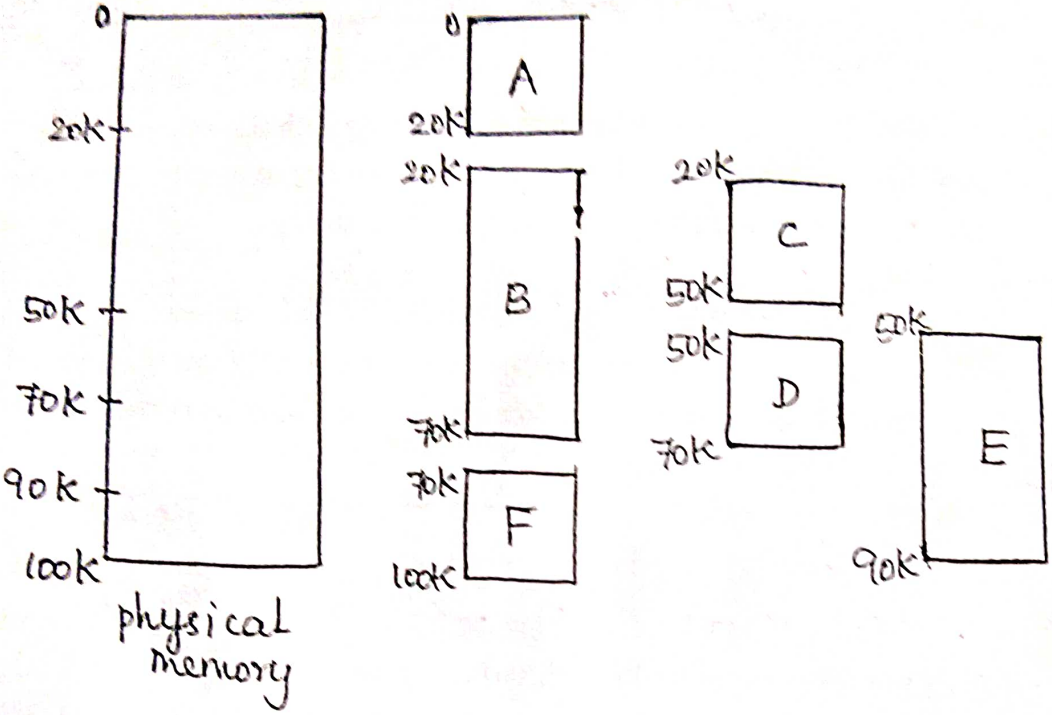


~~space=190kb~~

Total address
Space = 190 KB

e) Instead of using 190kb to hold the job's address space in memory, a maximum of any 100 k bytes are needed using overlay technique as follows:

Total address = 190K bytes



(b) Overlay assignment.

UNIVERSITY QUESTIONS

2 MARKS

1. State the uses of OS.
2. What is interrupt?
3. Define dynamic relocation.
4. What is roll-out?
5. Define process scheduling.
6. List out the functions of process scheduler.
7. State the functions of device management categories.
8. What is spooling?
9. Define file system.
10. What is kernel?
11. Define OS. Give example.
12. What do you mean by user?
13. What is called a degree of multiprogramming?
14. What is demand paged memory management?
15. What is deadly embrace or deadlock?
16. What is a race condition?
17. What are referred as SPOOL and DASD?
18. What is logical file system?
19. Define JCL.
20. What do you mean by scheduler?
21. Distinguish between contiguous and non contiguous allocation.
22. What is paging.
23. What is rotational delay?
24. State any two basic functions of file system.
25. What do you mean by 'throughput'?
26. What are semaphores?
27. Differentiate internal and external fragmentations.
28. What is compaction?
29. List any two page replacement algorithms.
30. What is the role of I/O scheduler?
31. What are the primary functions of physical file system?
32. List any two techniques for non-contiguous allocation of file system.

5 MARKS

1. Explain about memory management and processor management functions.
2. Short notes on single contiguous allocation.
3. Define interrupt. Explain different types of interrupt.
4. Write about i. Shared devices ii. Dedicated devices.
5. Explain any one non-contiguous allocation technique in detail.
6. Write about I/O Traffic Controller
7. Explain about wait and signal mechanism for process synchronization.
8. Explain about overlays with an example.
9. Write a note on process scheduling policies.
10. Write about inodes of UNIX.
11. Explain i. Time sharing system ii. Batch processing
12. Explain i. Round Robin policy ii. Shortest job first
13. What is fragmentation? What are its types? How does it occur? How can it be tackled?
14. Compare various page replacement strategies?

15. What are private and public key? How are they used in cryptography?
16. Discuss Access Control Mechanisms in context of data files.
17. Write a note on other views of OS.
18. Write short notes on semaphores P and V operations.
19. Explain techniques of device managements.
20. Write notes on I/O scheduler and i/O device handler.
21. Write short notes on Simple File System.
22. Describe about UNIX.
23. Classify and explain the various types of I/O channels.
24. Describe about swapping and overlays.
25. Outline the advantages and disadvantages of re-locatable partitioned allocation.
26. Describe the contents of PCB in detail.
27. Briefly explain Race condition.
28. Describe about spooling system.
29. Explain about any 4 UNIX commands with example.
30. Write a note on time sharing devices.

10 MARKS

1. Explain various process states and process state transition with diagram
2. Explain in detail about segmented memory management.
3. Explain the techniques for handling deadly embrace.
4. Explain Direct Access Storage Devices.
5. Explain about the general model of a file system.
6. Explain the evolution of OS from single batch processing system to today's standard OS.
7. Describe briefly device management functions.
8. Discuss various scheduling policies.
9. Explain about Banker's algorithm to avoid a deadlock. What are the problems in implementation?
10. Explain in detail about demand-paged memory management.
11. Explain I/O traffic controller with neat diagram.
12. Explain the criteria to be considered while determining job scheduling policies.
13. Explain physical file system in detail.
14. Describe about virtual devices.
15. Explain about designing and implementation of OS.

****