Unit: III
Lecture: 6
(Memory Management)
Memory Allocation Techniques (Part-III)

Non-contiguous Memory Allocation:

In non contiguous memory allocation, processes can be allocated anywhere in available space. Various non-contiguous memory management schemes are:

1. **Paging**
2. **Segmentation**

Paging: Paging is a memory-management scheme that permits the physical address space of a process to be noncontiguous.

Paging avoids external fragmentation and the need for compaction.

The basic method for implementing paging involves

- breaking physical memory into fixed-sized blocks called frames.
- and breaking logical memory into blocks of same size called pages.

When a process is to be executed, its pages are loaded into any available memory frames from their source.

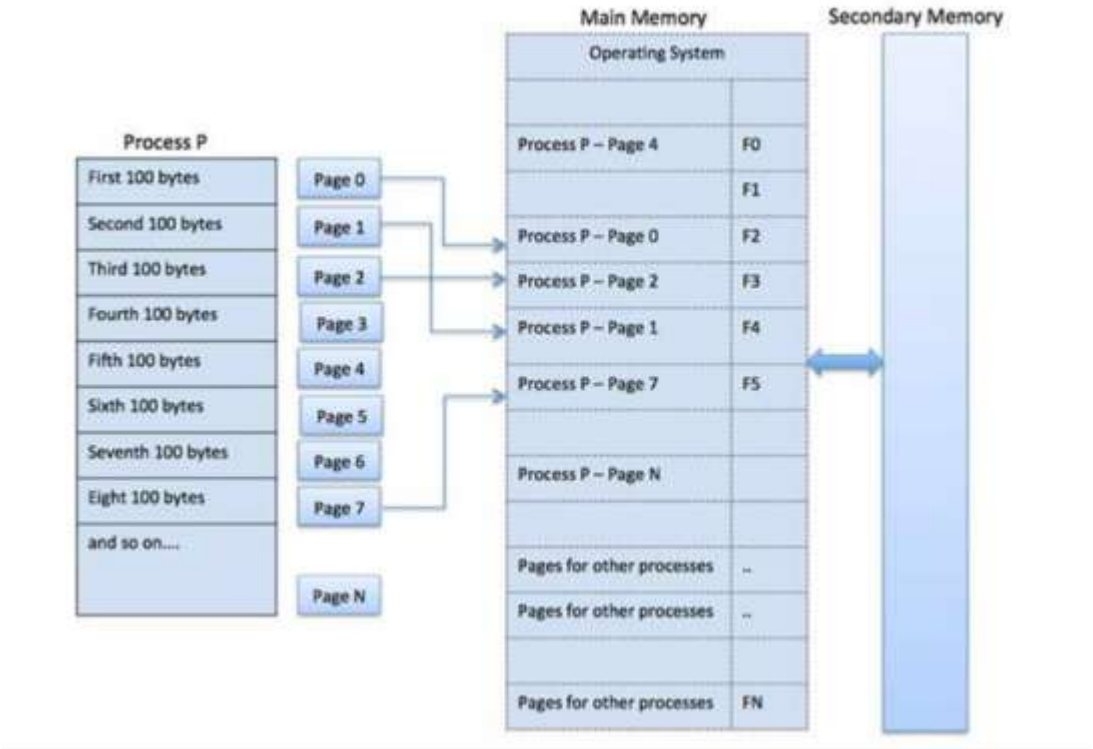


Fig: Paging Scheme

The hardware support for paging shown below:

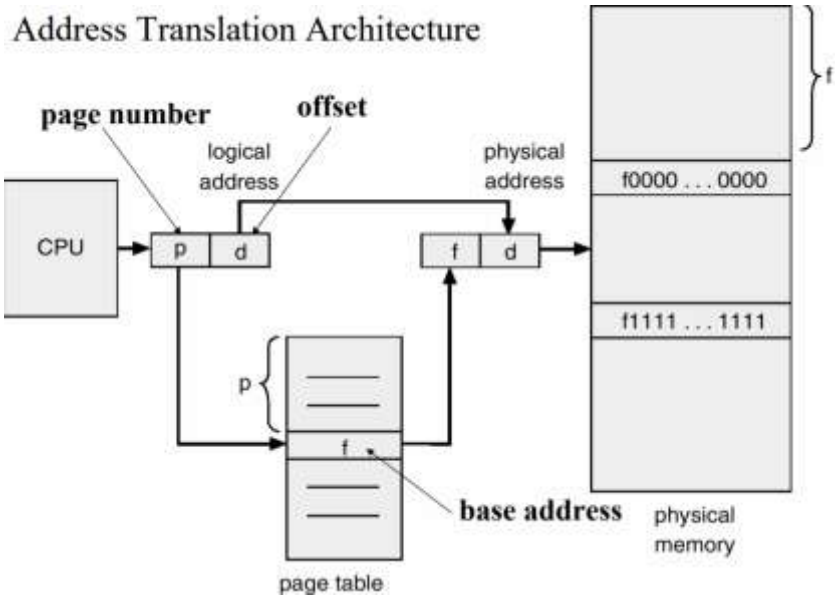


Fig: Paging Hardware

Every address generated by the CPU is divided into two parts:

- a page number (p): number of bits required to represent the pages in Logical address space.
- a page offset (d): number of bits required to represent particular word in a page or word number of a page.

i.e. Logical Address = Page Number (p) + page offset(d)

The page number (p) is used as an index into a page table.

The page table contains the base address of each page in physical memory.

This base address is combined with the page offset to define the physical memory address that is sent to the memory unit.

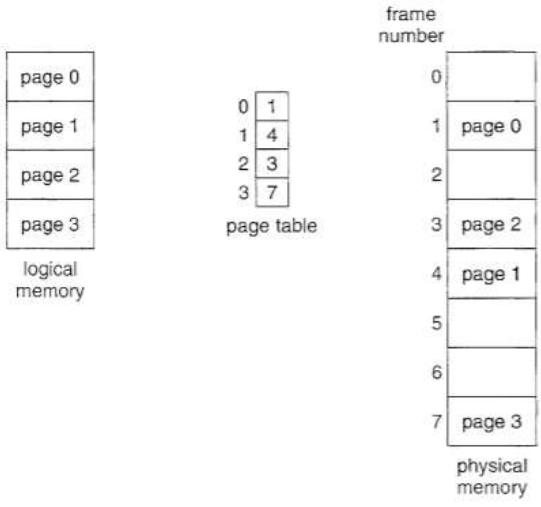


Fig: Paging model of logical and physical memory

Physical Address is also divided into:

- frame number(f): number of bits required to represent the frame of Physical Address Space.
- frame offset(d): number of bits required to represent particular word in a frame or word number of a frame.

i.e. Physical Address = Frame number(f) + frame offset (d)

The page size (like the frame size) is defined by the hardware.

and Page size = Frame size.

If the size of the logical address space is 2^m and a page size is 2^n addressing units (bytes or words), then (m-n) bits of logical address designate the page number and n designate the page offset.

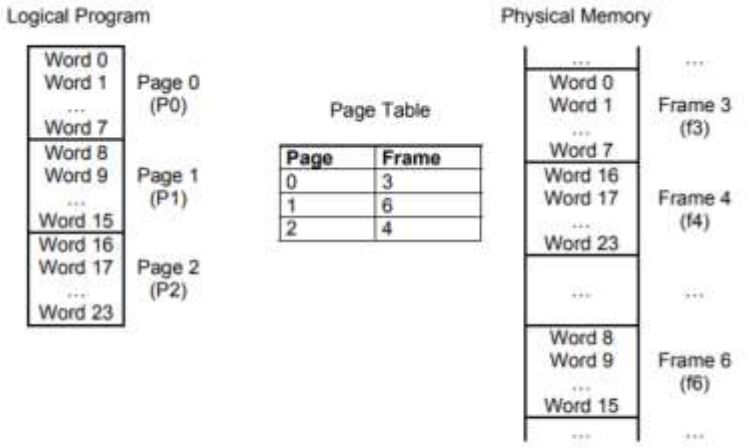
Consider the following information:

Page Size = 8 words

Physical Memory Size =128 words

Let a program of 3 pages P0, P1, P2 where P0 is in frame f3 P 1 is in frame f6 and P2 is in frame f4.

Thus, the above program in paging scheme will be mapped as shows below:



Program Line	Logical Address	Offset	Page Number	Frame Number	Physical Address
Word 0	00 000	000	00	011	011 000
Word 1	00 001	001	00	011	011 001
...	...	...	...	...	...
Word 7	00 111	111	00	011	011 111
Word 8	01 000	000	01	110	110 000
Word 9	01 001	001	01	110	110 001
...	...	...	...	...	...
Word 15	01 111	111	01	110	110 111
Word 16	10 000	000	10	100	100 000
Word 17	10 001	001	10	100	100 001
...					
Word 23	10 111	111	10	100	100 111

Now, let's **consider another example as shown below**:

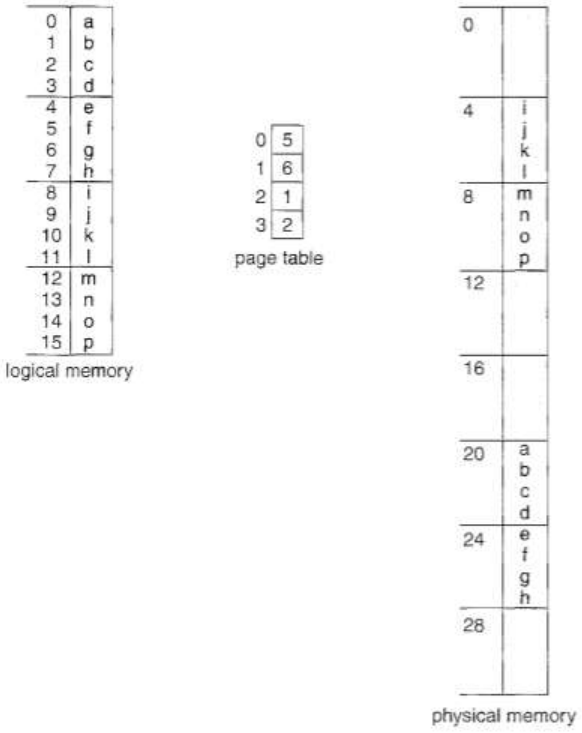


Fig: Paging example for a 32 byte memory with 4 byte pages

Consider a Process with 4 pages (P0, P1, P2, P3) as shown in above fig in logical memory.

Let **page size be of 4 bytes and physical memory is of 32 bytes.**

As page size is 4 bytes → n =2 i.e. page offset = 2.

Now lets see how user's view of memory can be mapped into the physical memory in this example.

Logical address 0 maps to 20 in the physical memory as shown in the diagram because:

From the Page table, we can see Page 0 is in frame 5.

Now, Physical Memory Address is: Frame Number + frame offset (page offset)

Here as mentioned above page offset is represented by 2 bits i.e. page offset will range from 00,01,10,11,

Thus, Logical address 0-4 will be mapped to physical address:

10100

10101

10110

10111

which is equivalent to 20-23 as shown in the diagram too.

Similarly, logical address 4 maps to physical address 24 and logical address 13 maps to physical address 9.

or we can say: Physical Address = frame Number * Page Size + page offset.

e.g. physical address for logical address 0 is = 5*4+0 =20

physical address for logical address 1 is = 5*4+1 = 21

physical address for logical address 13 is = 2*4+1=9