Laboratory work No. 5

Analysis of paging memory management scheme

1. **Aim of the work**: analyse paging memory management scheme.

2. The Theoretical Part

Paging is a memory management scheme that permits the physical address space of a process to be non-contiguous. Paging avoids the considerable problem of fitting memory chunks of varying sizes onto the backing store. From this problem suffered most memory management schemes before the introduction of paging.

The basic method for implementing paging involves breaking physical memory into fixes sized blocks called frames and breaking logical memory into blocks of the same size called pages. When a process is to be executed, its pages are loaded into any available memory frames from the backing store. The backing store is divided into fixed sized blocks that are of the same size as the memory frames. 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. The paging model of memory is shown in figure 1.

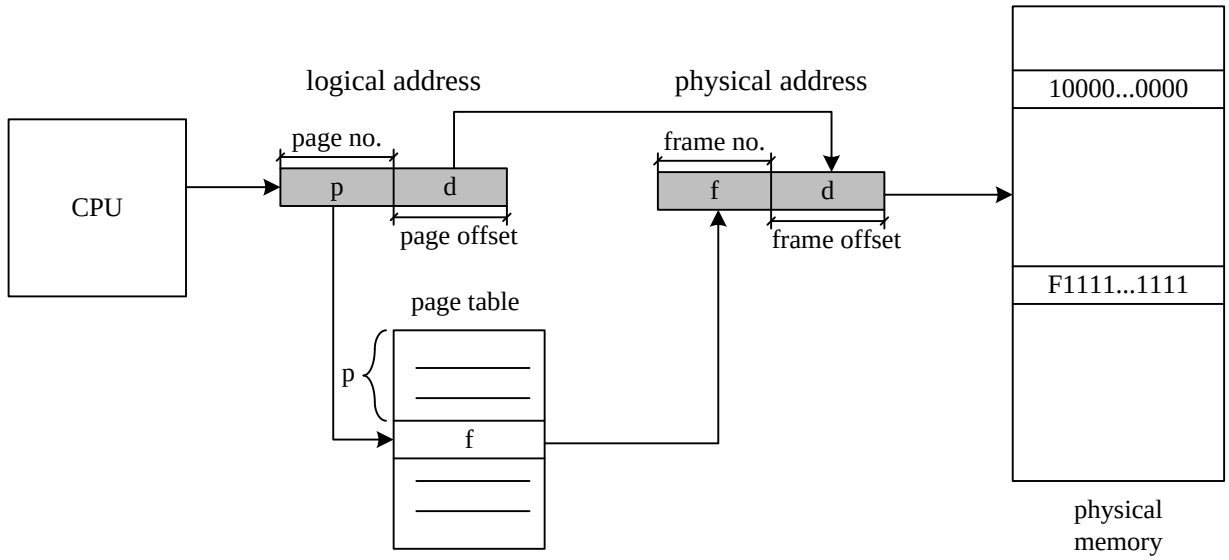

Fig 1. Paging hardware

Every address generated by the CPU is divided into two parts: page number – p and a page offset – d. The page number is used as an index into a page table. Logical address structure is shown in figure 2. If the size of logical address space is 2^m, and a page size is 2^n (bytes or words), then the high order m – n bits of a logical address designate the page number, and the n lower order bits designate the page offset.

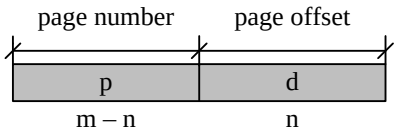

Fig 2. Logical address structure

The page size (like the frame size) is defined by the hardware. The size of a page is typically a power of 2, varying between 512 bytes and 16 MB per page, depending on the computer architecture.

In figure 3 shown example of memory mapped into physical memory, where page size is 4 bytes and a physical memory of 32 bytes (8 frames). Logical address 0 is page 0, offset 0. Indexing into the page table, we find that page 0 is in frame 5. Thus, logical address 0 maps to physical 20 address ((5·4)+0). Logical address 3 (page 0, offset 3) maps to physical 23 address ((5·4)+3). Logical address 4 is page 1, and offset – 0. From the page table, we find that page 1 is in frame 3, thus, physical address is 12 ((3·4)+0). Page 2 mapped into 0 frame, 3 page into 7 frame.

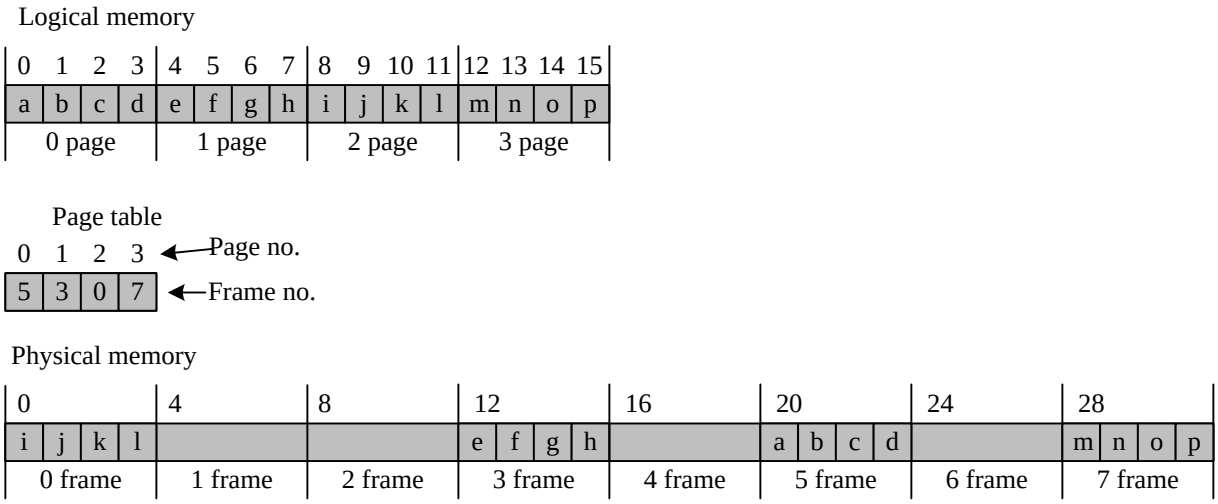

Fig 3. Paging model of logical and physical memory

3. Work task

Write code which illustrates paging memory management scheme. It must be possible to change logical and physical memories size (the exact size will be told during laboratory work), page size. It must be possible to change number of processes and their sizes.

Must be calculated and displayed: number of pages occupied by each process, page table, map pages to physical memory, list of non occupied frames.

Literature

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. Applied Operating Systems Concepts. 2000, 840 p. ISBN 0-471-36508-4.