COMPUTER ARCHITECTURE AND SOFTWARE EXECUTION PROCESS

MEMORY MANAGEMENT

🎓 Bachelor in Artificial Intelligence, Data and Management Sciences
🏛 CentraleSupelec and ESSEC Business School - 2023/2024

Idir AIT SADOUNE
idir.aitsadoune@centralesupelec.fr

OUTLINE

- Classification
- Cache memory
- Memory Management

Back to the outline - Back to the begin

OUTLINE

- Classification
- Cache memory
- Memory Management

Back to the outline - Back to the begin

LOCALISATION

- Internal processor memory
- Main memory
- External memory

PHYSICAL CHARACTERISTICS

- Volatile / non-volatile
- Read only / read and write
- Destructive / non-destructive reading
- Erasable / non-erasable

ACCESS METHOD

- Random access
- FIFO or LIFO access
- Associative access
- Direct access
- Sequential access

TWO IMPORTANT ACRONYMS

1. RAM (Random Access Memory)
    - read/write access
    - random access
    - volatile

2. ROM (Read-Only Memory)
    - read-only access
    - random access
    - non-volatile

OUTLINE

- Classification
- Cache memory
- Memory Management

Back to the outline - Back to the begin

TWO OBSERVATIONS

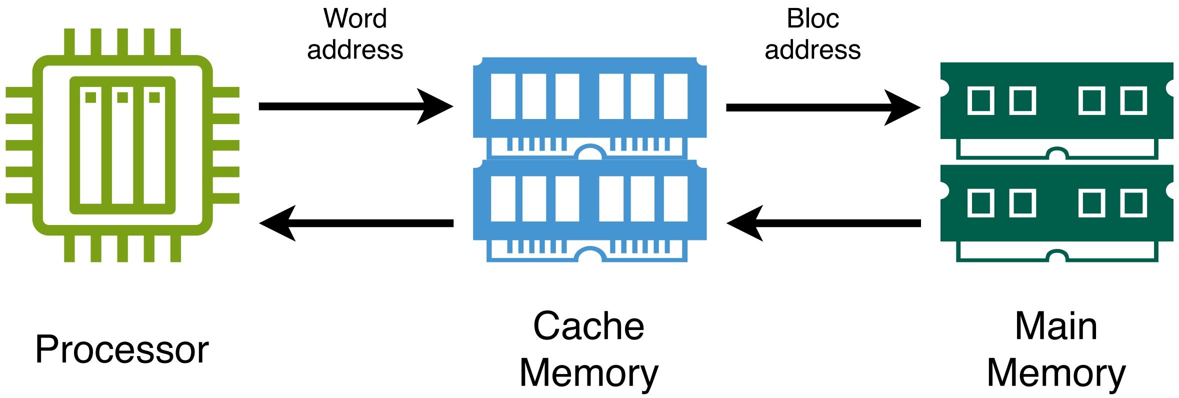
1. **Temporal Locality**
    - When a processor searches for a word in memory, it is highly probable that it will need the same word soon after.

   ➠ Keep recently used words in quick memory.

2. **Spatial Locality**
    - When a processor searches for a word in memory, it is highly probable to need a neighbouring word shortly after.

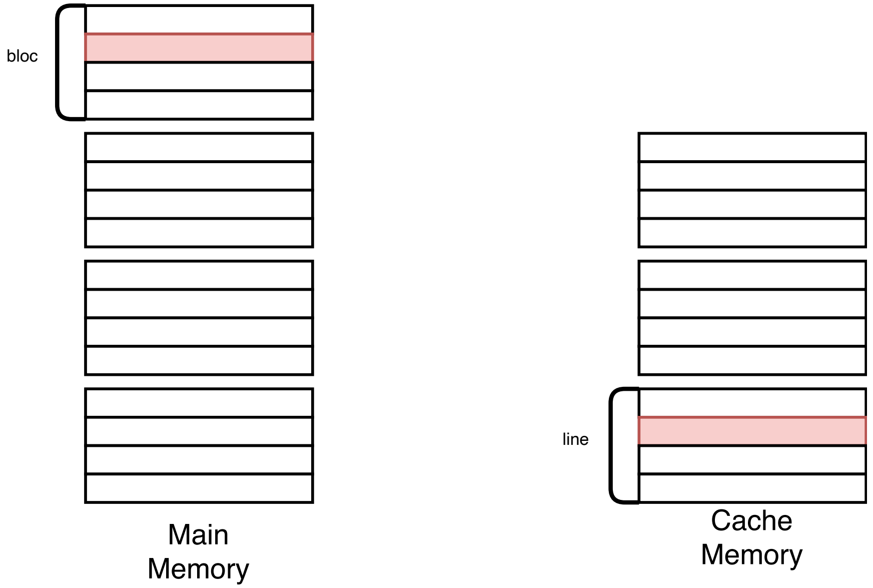
   ➠ Do not store an isolated word in fast memory, but a block of contiguous words.

CACHE PRINCIPLE

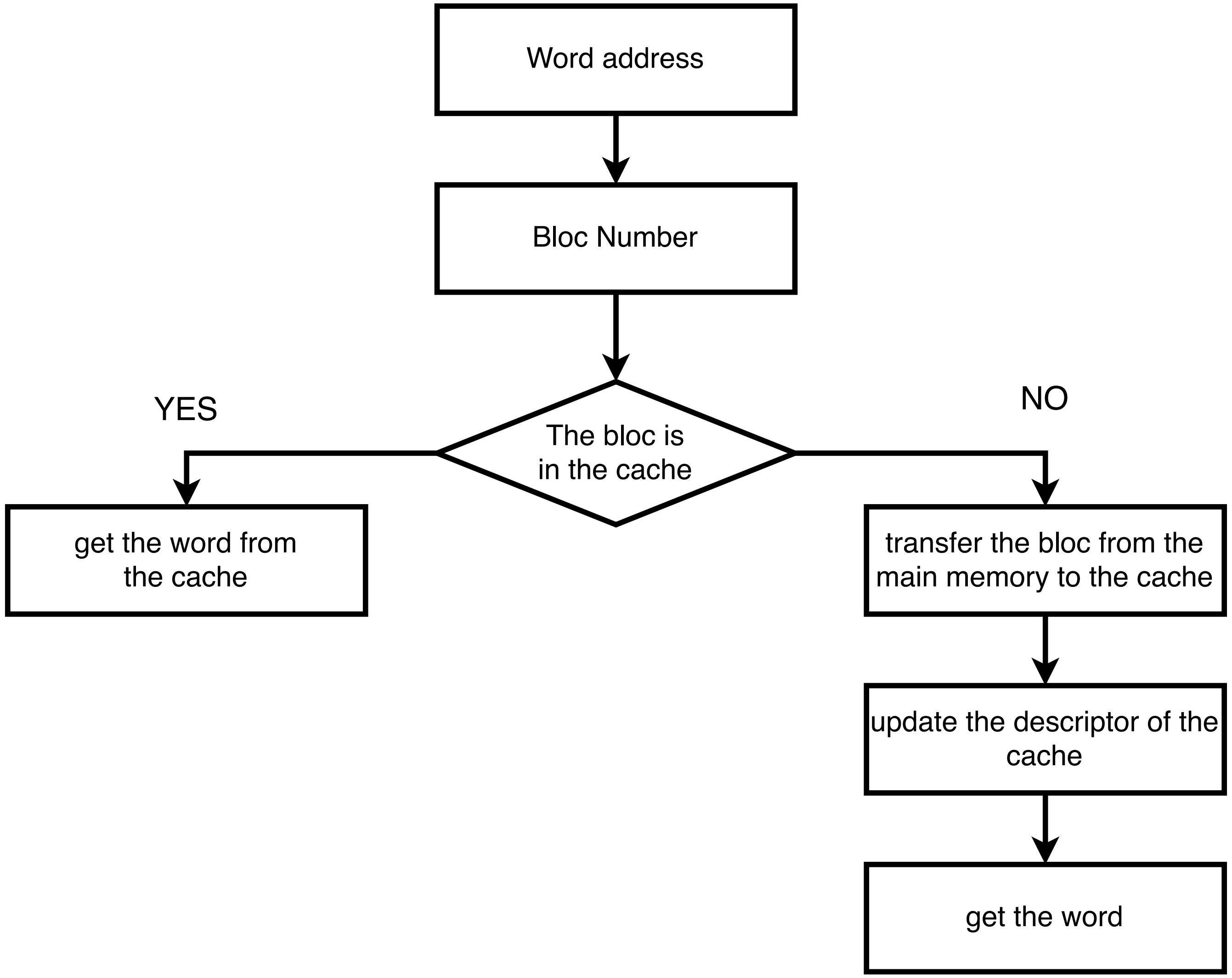


CACHE PRINCIPLE

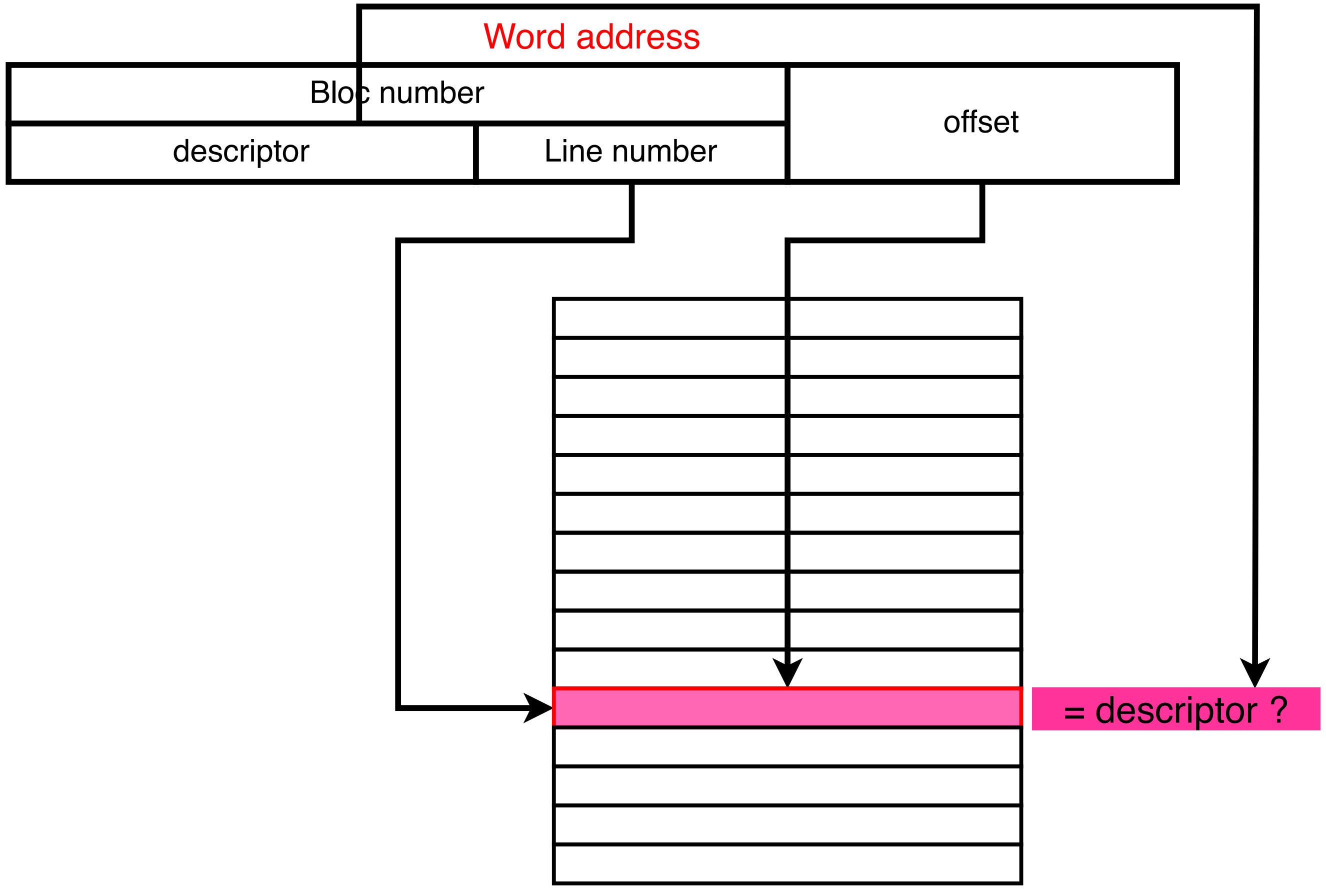
- The main memory is virtually divided into blocks containing $M = 2^m$ words.
- A cache is organized in N lines; each line can contain one block.
- Each line has a descriptor allowing to know which block it contains.



HOW CACHE WORKS


Word address
Bloc Number
The bloc is in the cache
YES
NO
get the word from the cache
transfer the bloc from the main memory to the cache
update the descriptor of the cache
get the word

HOW CACHE WORKS



OUTLINE

- Classification
- Cache memory
- Memory Management

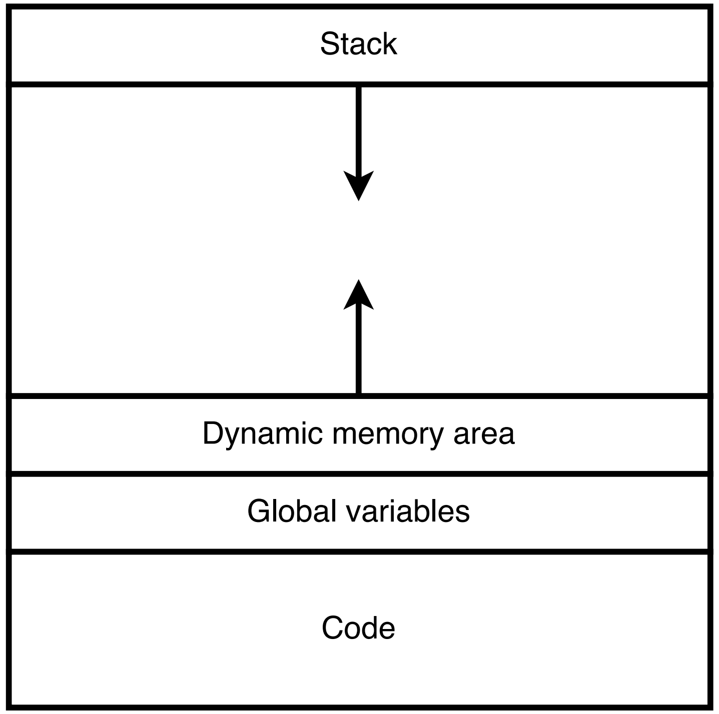
Back to the outline - Back to the begin

SOME QUESTIONS

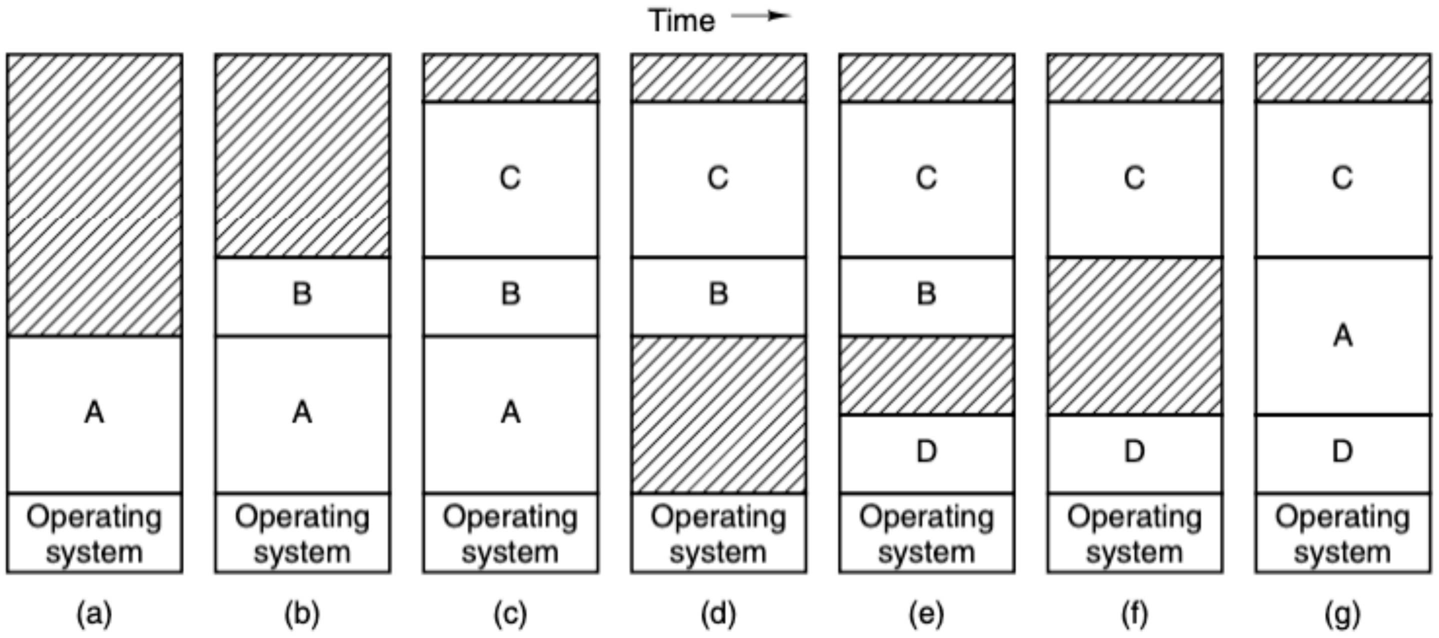
- Each process must use a separate memory area (address space) for **security reasons**.
  - What mechanism for allocating this space?
  - How to ensure the protection of this area?
  - How can we ensure the transparency of the position of this space concerning a program?

ADDRESS SPACE USAGE

- What does a process's memory space contain?
  - Code (known size)
  - Global variables (known size)
  - Stack (unknown size)
  - Dynamic memory area (unknown size)



MULTIPROGRAMMING WITH PARTITIONS

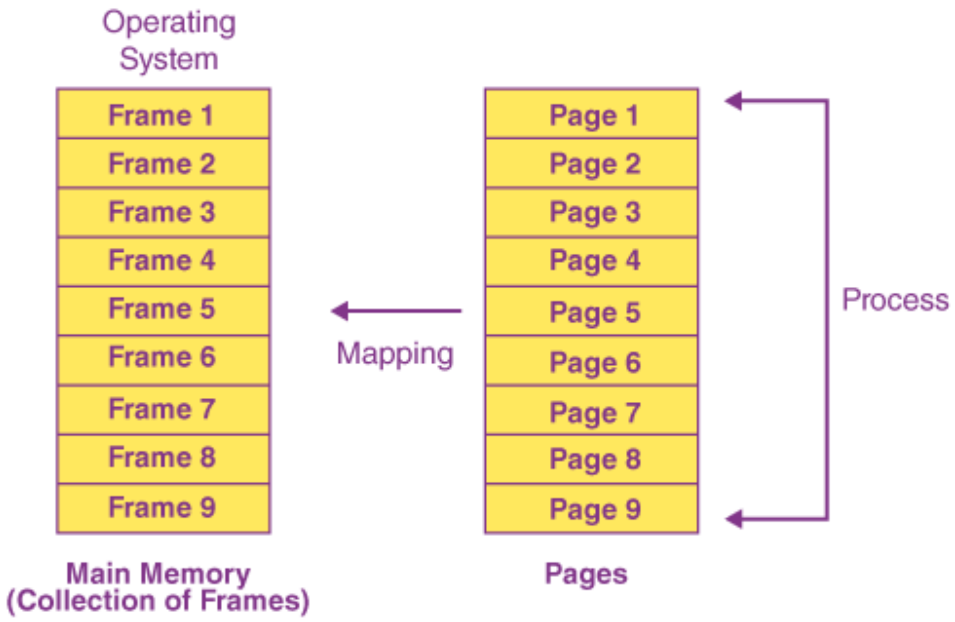


MULTIPROGRAMMING WITH PARTITIONS

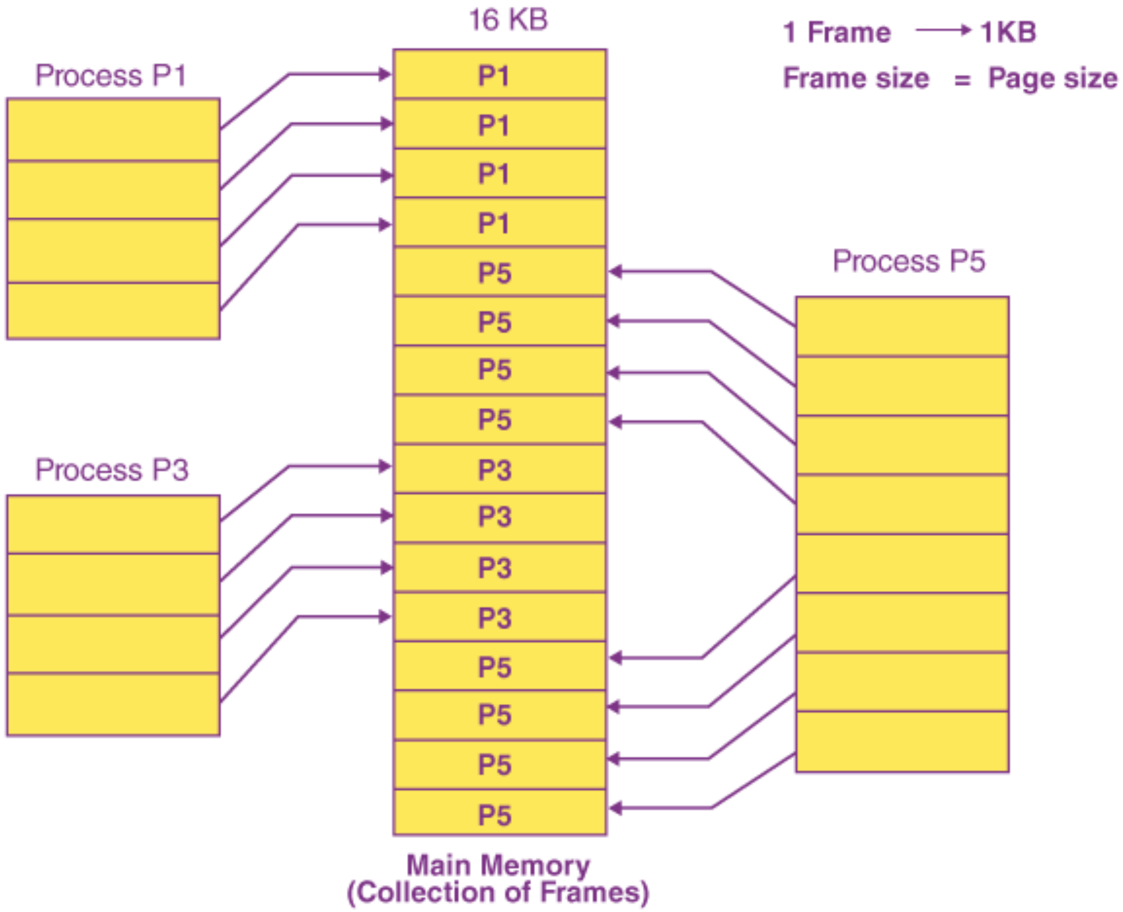
- Pros
  - ✔ Material simplicity
  - ✔ Transparency for programs
  - ✔ Checking the validity of addresses
- Cons
  - ✘ Fragmentation
  - ✘ Fixed size of memory spaces

MULTIPROGRAMMING WITH PAGING

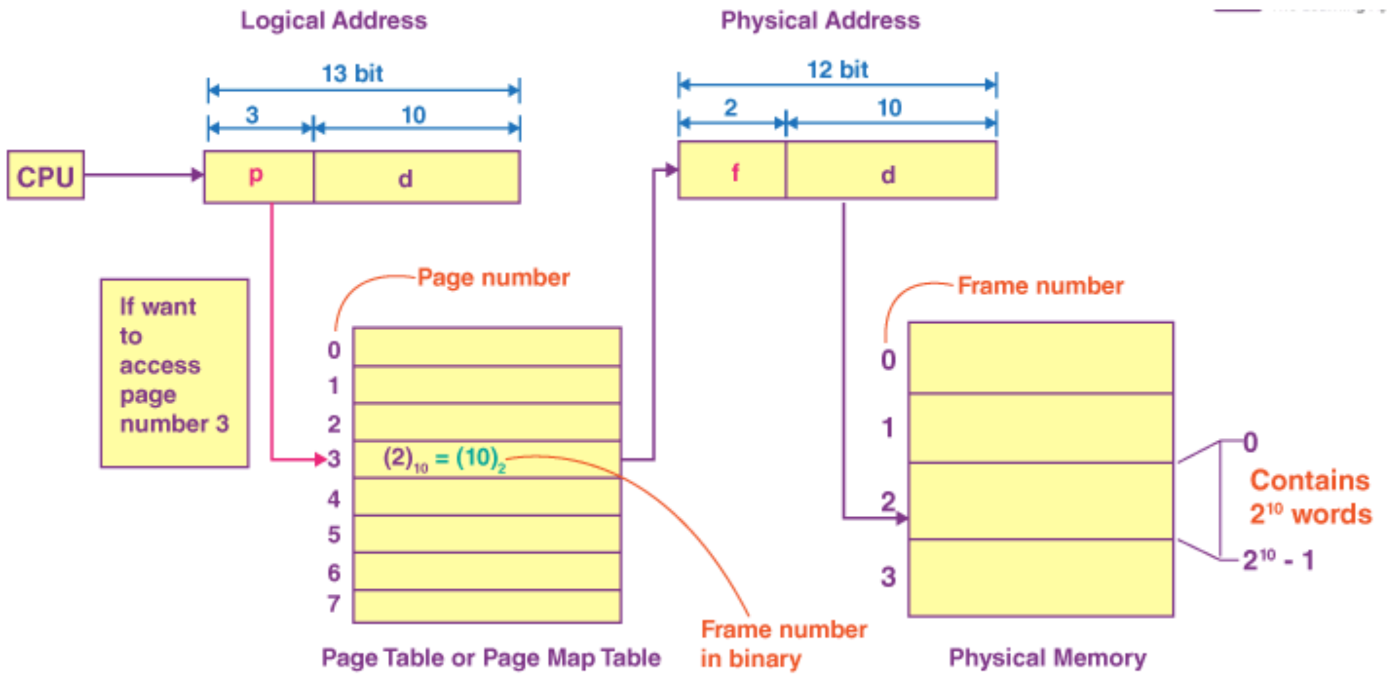
- The Main Memory is virtually divided into **blocks** (**frames**) allocated independently to processes.
- The address space of a process is divided into **pages**.
  - the size of a page is the same as the block size.



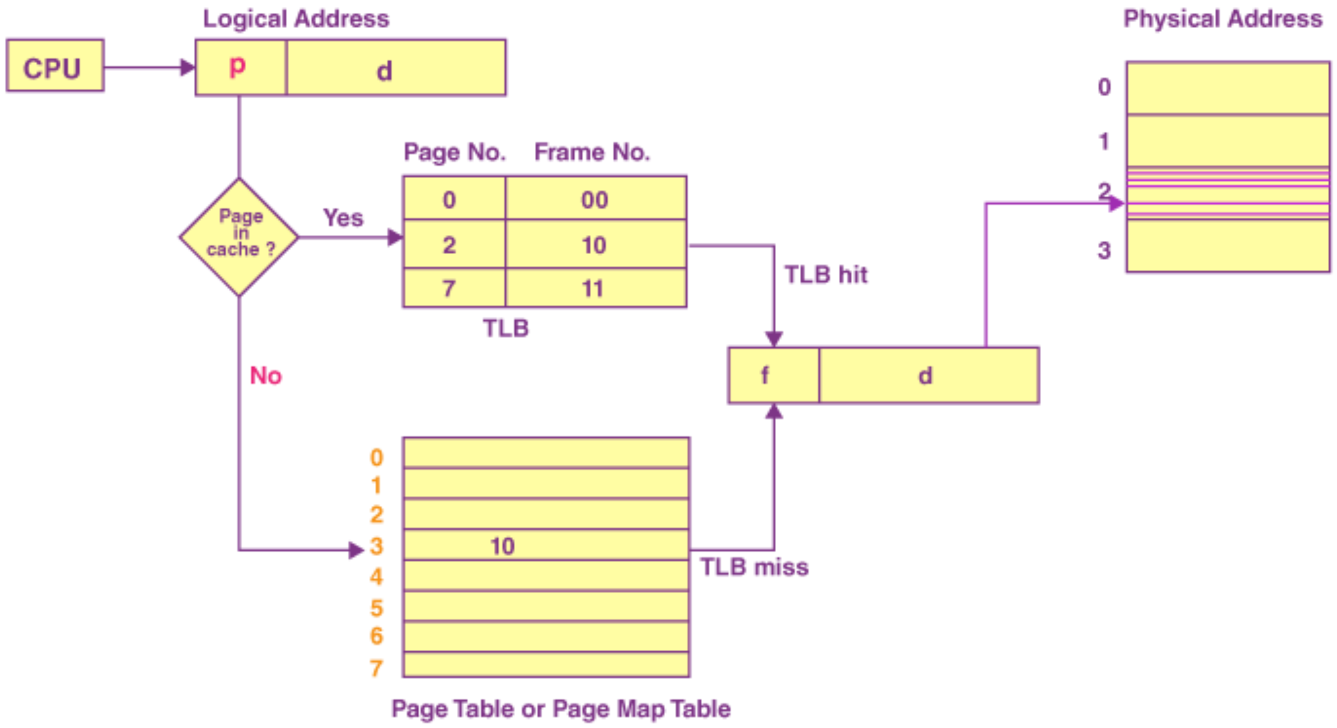
MULTIPROGRAMMING WITH PAGING



MULTIPROGRAMMING WITH PAGING



MULTIPROGRAMMING WITH VIRTUAL MEMORY



- Space of usable pages larger than space of physical memory
- The pages are either in the main memory or on the hard disk.

THANK YOU

Back to the begin - Back to the outline