Computer Fundamentals
Logical Address Space (1/2)

Grau en Intel·ligència Artificial

Xavier Martorell

Xavi Verdú

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC)

2021-2022 Q1

Creative Commons License

Table of Contents

- This lesson consists of three main parts
  1) **Basic Concepts**
  2) Libraries and OS suport

Concepts

- **Processor** address space
  - Subset of addresses that the processor can issue (depends on the bus of addresses)
- **Process** is a program that has been launch to be executed
  - The Operating System (OS) assigns resources to processes (e.g. CPU access, memory, etc)
- Memory addresses generated by the CPU are **logical addresses**
- **Process logical address** space
  - Subset of logical addresses that a process can reference (OS decides what are those valid addresses for each process)
- Memory addresses that reach the memory are **physical addresses**
- **Process physical address** space
  - Subset of physical addresses associated to the process logical address space (decided by the OS kernel too)

Logical addresses vs Physical addresses

- What if they are the same?
  - Fixed: logical address space == physical address space

- What if they are different?
  - Translation required: It can be done at different moments
    - Option1, During program loading:
      - OS decides where to place the process in memory and translate references at program loading
    - Option 2, During program execution: each issued reference is translated at runtime (this is the normal behavior in current systems)
      - Collaboration between HW and OS
        - HW offers the translation mechanism: **Memory Management Unit (MMU)**
        - OS configures it

Multiprogrammed Systems

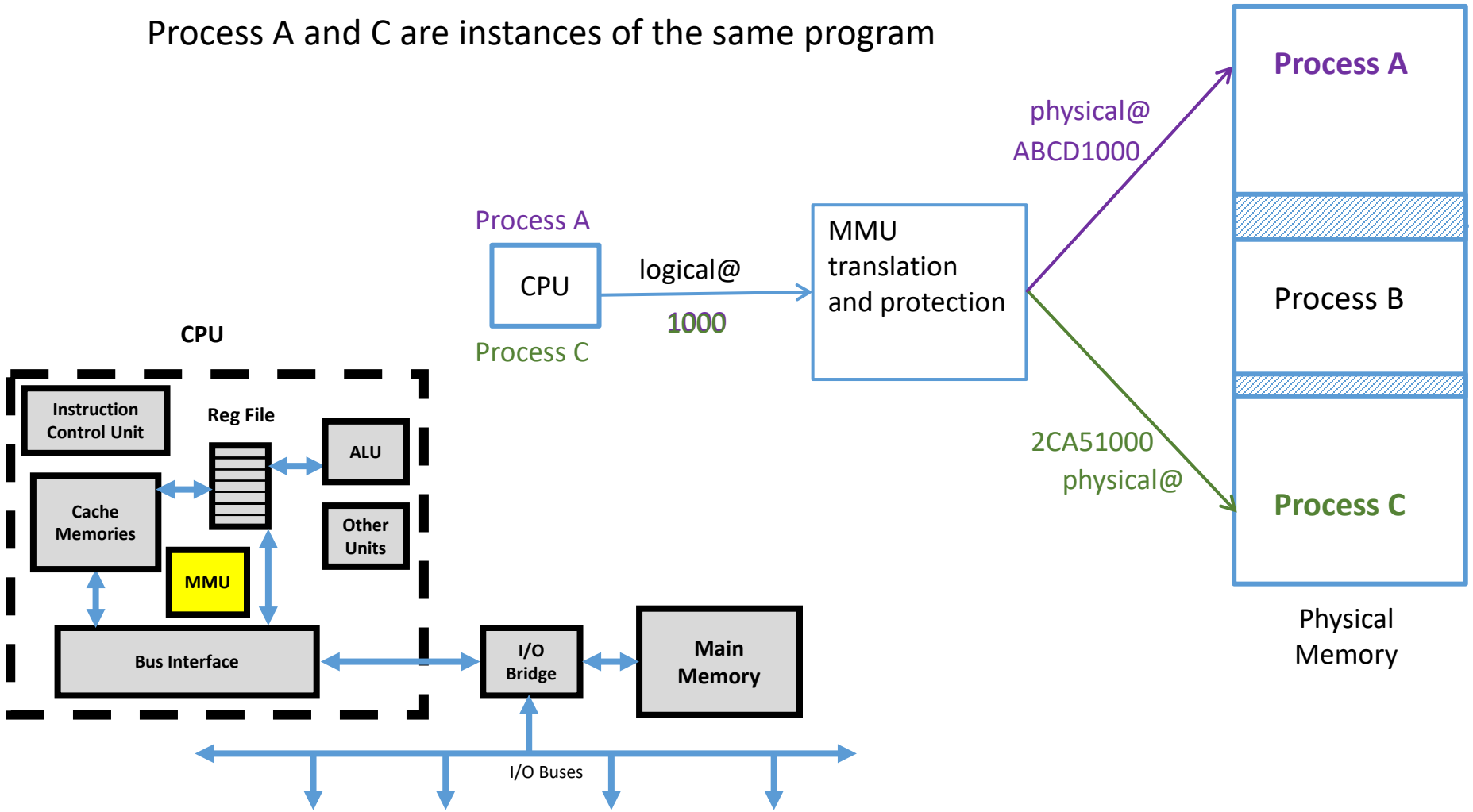
- Several programs loaded simultaneously in physical memory
- Ease concurrent execution and simplify context switch mechanism
  - 1 process on CPU but **N processes in physical memory**
  - When the process that uses the CPU is a different one it is not necessary to load again the contents
- OS must guarantee **physical memory protection**
  - Each process can access only the physical memory that it gets assigned
  - **Collaboration between HW and OS**
    - MMU implements the mechanism to detect illegal accesses
    - OS configures MMU
- OS must update MMU according to any change in the running processes:
  - When performing a context switch, OS updates MMU with the information of the process that gets assigned the CPU
  - If a process address space grows
  - etc….

Memory Management Unit (MMU)

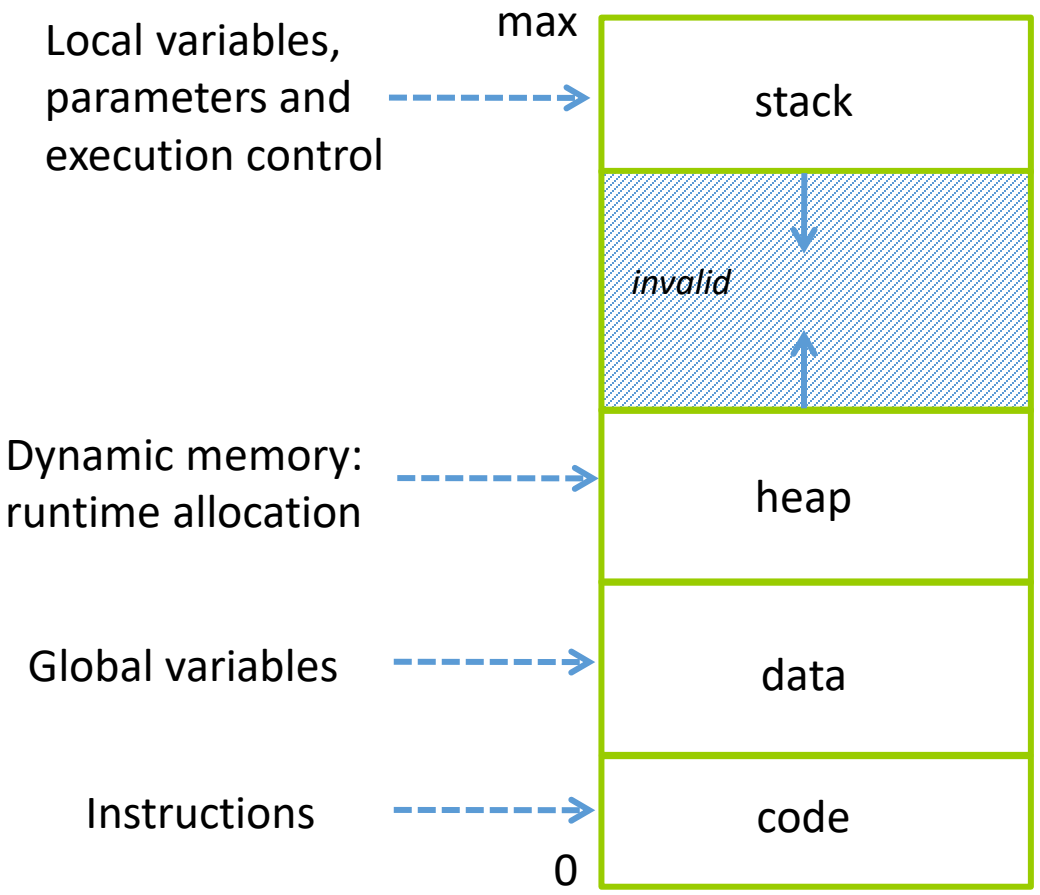
- MMU: HW component which, at least, offers **address translation and memory access protection**. It can also support other management tasks.
- **OS is responsible for configuring the MMU with the correct address translation values for the current process in execution**
  - What logical @ are valid and what are their corresponding physical @
  - Guarantees that each process gets assigned only its own physical @
- **HW support to translation and protection between processes**
  - MMU receives a logical @ and translates it to the corresponding physical @ using its data structures
    - It throws an exception to the OS if the logical address is not marked as valid or if it has not associated a physical address
  - OS manage the exception according to the situation: **Segmentation Fault!!!**
    - **Exception:** unvoluntary issue triggered by the execution of an instruction

Multiprogrammed Systems

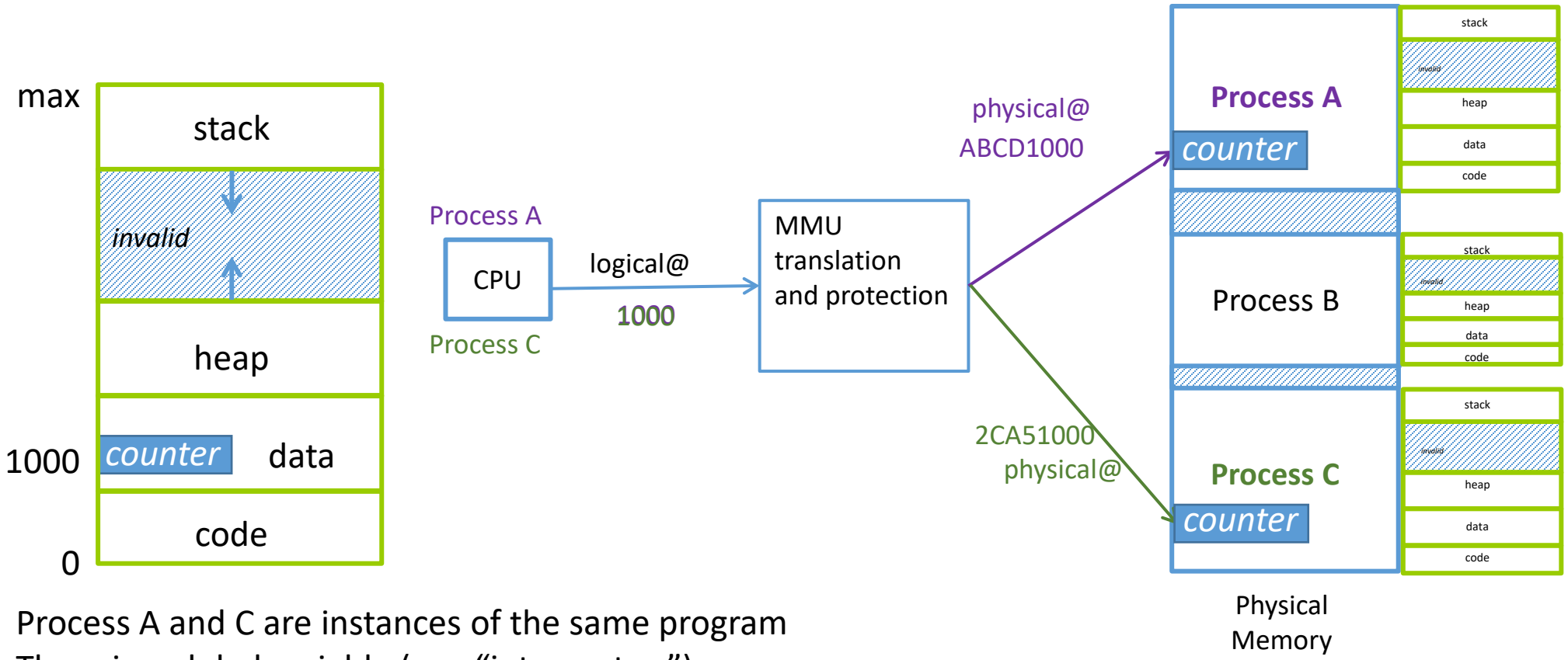
Process A and C are instances of the same program



Process Logical Address Space



Combining Views



Process A and C are instances of the same program
There is a global variable (e.g. “int counter;”)
- The compiler has assigned the logical @ 1000
    - Since it is an integer, the four bytes of the variable go from @1000 to @1003

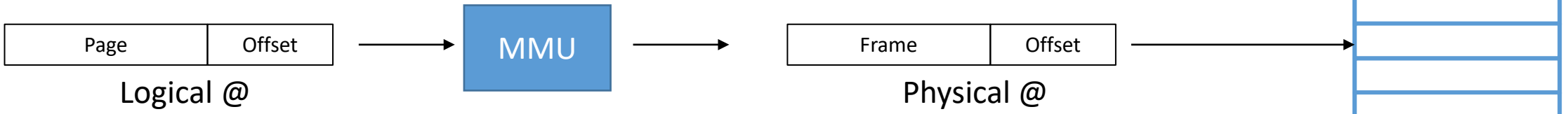
Memory Assignment

- **Contiguous assignment**
  - Physical address space is contiguous
    - The whole process is loaded on a partition which is selected at loading time
  - It is not flexible and complicates to apply optimizations (as, for example, on-demand loading)
- **Non-contiguous assignment**
  - Physical address space is not contiguous
  - Flexible
  - Increases granularity to the memory management of a process
  - Increases complexity of OS and MMU
- Based on
  - **Paging** (fixed partitions)
  - **Segmentation** (variable partitions)
  - Combined schemes
    - For example, paged segmentation

Memory Assignment Issue: Fragmentation

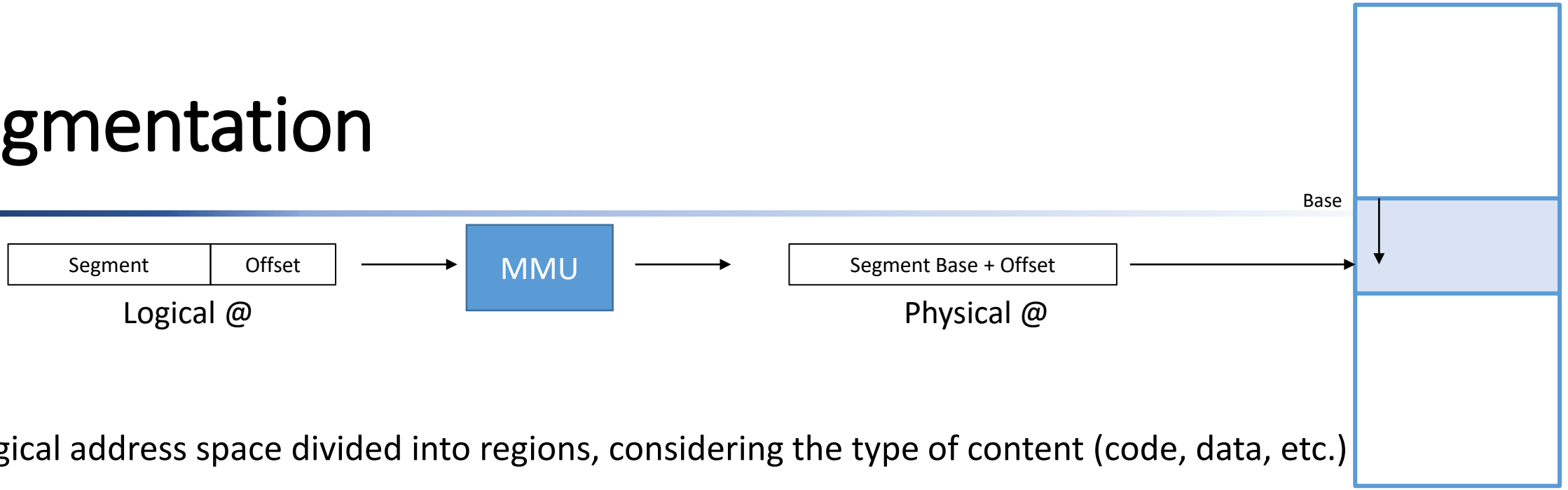
- **Fragmentation problem:** when it is not possible to satisfy a given memory request although the system has enough memory to do it. There is free memory, but it cannot be assigned to a process
  - Also mentioned in disk management
- **Internal fragmentation**: memory assigned (to a process) that is not going to be used
- **External fragmentation**: free memory that cannot be used to satisfy any memory request because it is not contiguous and large enough for the requirements
  - It can be avoided compacting the free memory. It is necessary the system to support address translation at runtime.

Paging



- Logical address space is divided into fixed size partitions: **pages**
- Physical memory is divided into partitions of the same size: **frames**
- Assignment
  - For each page look for a free frame
    - List of free frames
  - Can cause internal fragmentation
- The frames assigned to a process go back to the free frames list when the process ends the execution
- **Page: working unit of the OS**
  - Facilitates on-demand loading
  - Enables page-level protection
  - Usually, a page belongs to just one memory region to match region protection requirements (code/data/heap/stack)
- Can cause internal fragmentation

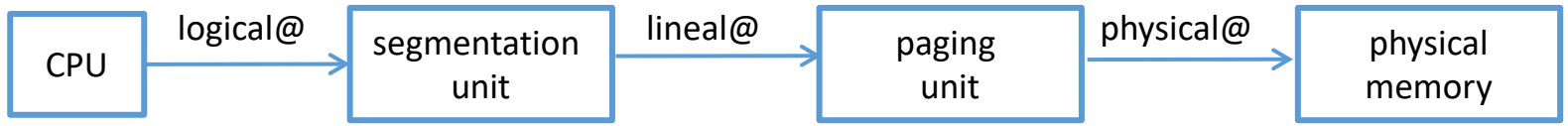
Segmentation



- Logical address space divided into regions, considering the type of content (code, data, etc.)
- Logical address space divided into variable size partitions **(segments)**, that fit the size that is really needed
  - At least 3 segments: one for code, one for stack and one for data
  - References to memory are composed of segment and offset
- Assignment: for each segment in a process
  - Look for a partition big enough to hold the segment
  - Possible policies: first fit, best fit, worst fit
  - Select from the partition just the amount of memory needed to hold the segment and the rest of the partition is kept in a free partitions list
- Can cause external fragmentation

Combined Scheme

- Paged Segmentation: 2 step translation



  - Logical address space is divided into segments
    - Code, data, heap, stack
  - Segments are divided into pages
    - Segment size is multiple of page size
    - Page is OS working unit

Bibliography

- Computer Systems – A Programmer's perspective (3rd Edition)
  - Randal E. Bryant, David R. O'Hallaron, Person Education Limited, 2016
  - https://discovery.upc.edu/permalink/34CSUC_UPC/11q3oqt/alma991004062589706711
    - Chapter 9

- Computer Organization and Design (5th Edition)
  - D. Patterson, J. Hennessy, and P. Alexander
  - http://cataleg.upc.edu/record=b1431482~S1*cat
    - Several Chapters