Operating Systems

(Memory management: Segmentation)
Slides Set #15

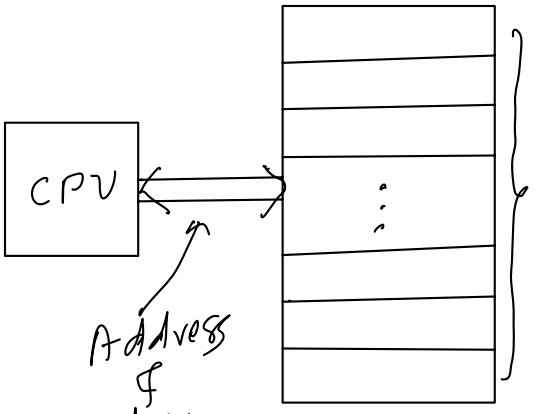
By Prof K R Chowdhary

JNV University

January 8, 2024

Memory Allocation

- One of the simplest methods for allocating memory is to divide memory into several **fixed-sized** partitions.
- In the **variable-partition** scheme, the operating system keeps a table indicating which parts of memory are available and which are occupied.
- At any given time, then, we have a list of available block sizes and an input queue.
- In general, the memory blocks available comprise a set of holes of various sizes scattered throughout memory.
- There are many solutions to this problem. The **first-fit**, **best-fit**, and **worst-fit** strategies

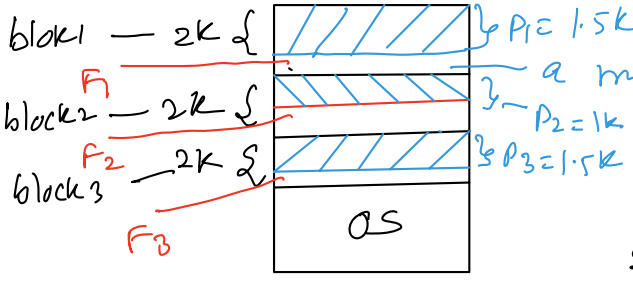


Locations of memory are like array, they store program & data

→ Function of memory management unit (MMU) is to allocate memory to programs and data, when progs are run.

→ memory allocation table keeps the information about which memory parts are available and which are allocated.

→ Consider a situation: the memory is divided into fixed partitions of 2K, and a prog P1 of size 1.5K is loaded into it.

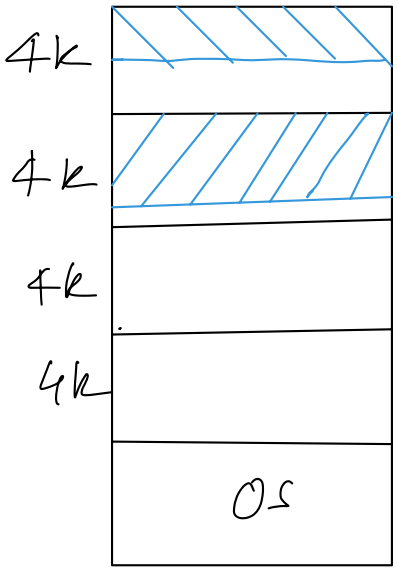


a memory fragment of 0.5K is created called Internal fragment as it is internal to the block of 2K.

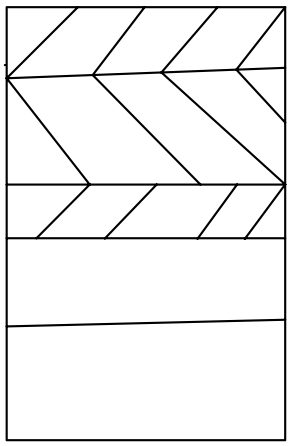
Say, a new prog. $P_2 = 1K$ is to be loaded now

→ For block 2, the F_1 is External Fragment (also called hole)

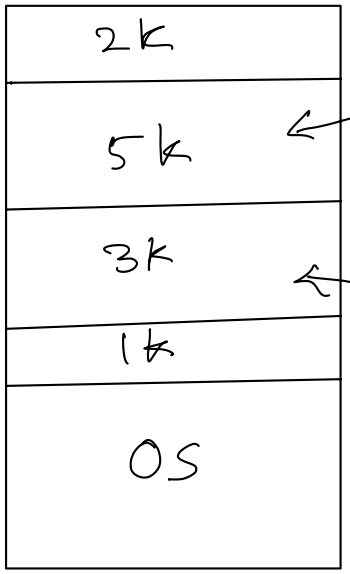
→ " " 2, " F_2 " Internal "



Fixed partition allocation (creates holes)



variable partition allocation, Now, if P2 completes, and new prog P4 of 1k is loaded, it will create a hole of 1k.



if prog. of 2.5K is loaded here, it is first-fit method.

if the prog. 2.5 K is instead loaded here, it is called best-fit method.

If largest partition is allocated to a program, it is called worst-Fit method.

Fragmentation

- Both the first-fit and best-fit strategies for memory allocation suffer from **external fragmentation**.
- Depending on the total amount of memory storage and the average process size, external fragmentation may be a minor or a major problem.
- Memory fragmentation can be **internal** as well as **external**.
- One solution to the problem of external fragmentation is **compaction**.

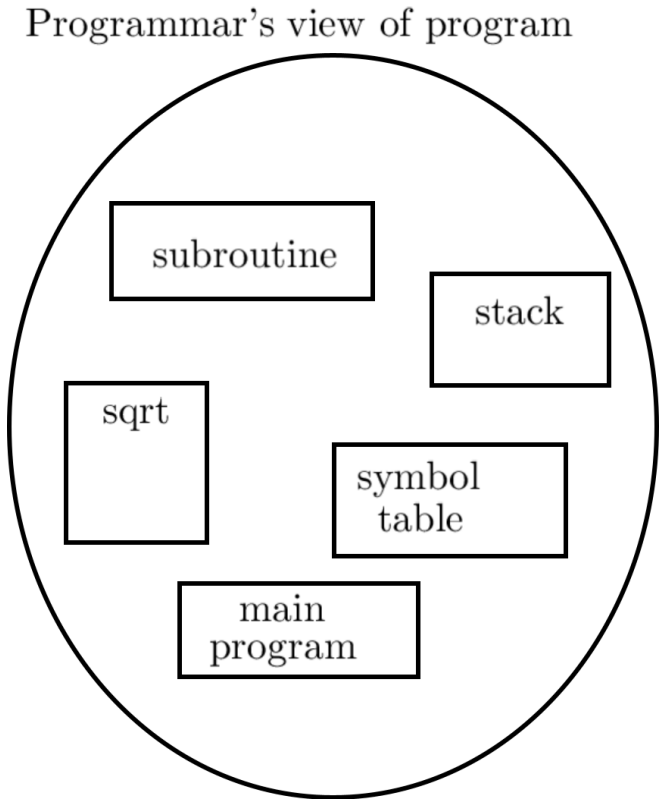
need to be done at run time, given that all fragments are pushed to either beginning or end

Segmentation

Another possible solution to the external-fragmentation problem is to permit the logical address space of the processes to be **noncontiguous**.

Important: **User's view** of memory is not the same as the actual physical memory.

- **Basic Method:** Do programmers think of memory as a linear array of bytes?,
- Programmer thinks of it as a main program with a set of methods, procedures, or functions.



- Segmentation scheme supports programmer view of memory.

→ A user considers the memory as a linear array of bytes of memory locations that contains either instructions or data. This is user's view of memory.

Fact: memory is divided into prog main(), its functions, stack, library Functions etc. Each having starting address and ending address which depends on where the program is loaded at run time.

→ A programmer considers the memory as different partitions of variable sizes, having some data, some programs, stack, functions etc.

→ The segmentation method supports the programmer's view of memory allocation.

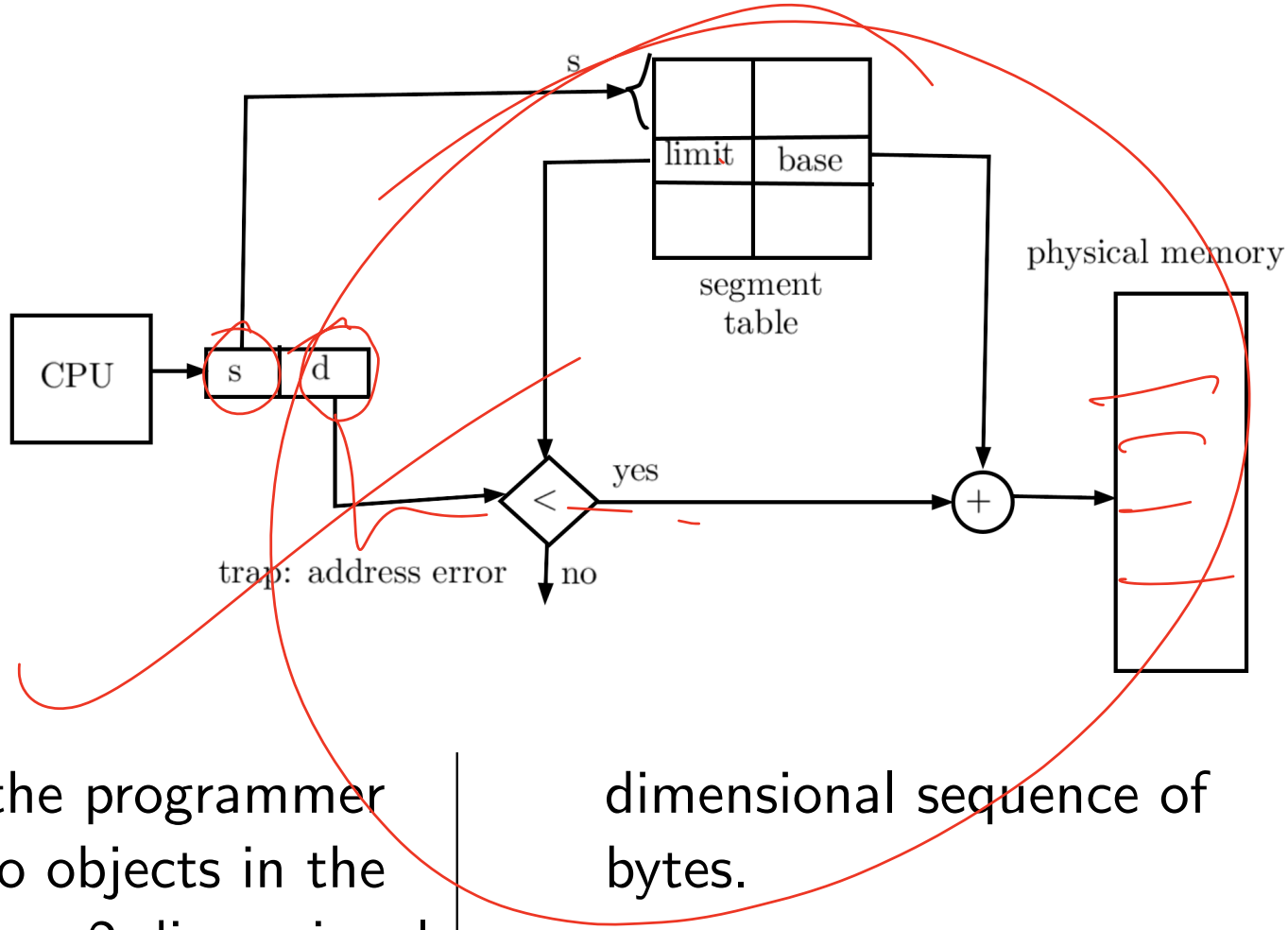
Segmentation....

- Each segment has a name and a length. The addresses specify both the segment name and the offset within the segment.

  <*segm_no.*, *offset*>.

  A 'C' compiler might create separate segments for the following:
  - The code,
  - Global variables,
  - The heap – from which memory is allocated,
  - The stacks used by each thread,
  - The standard C library.

Segmentation Hardware



- Although the programmer can refer to objects in the program by a 2-dimensional address, the actual physical memory is still a 1 dimensional sequence of bytes.
- The use of a segment table is illustrated in Figure