Virtual Memory

- Virtual memory is an alternate set of memory addresses.
- Programs use these virtual addresses rather than real addresses to store instructions and data.
- When the program is actually executed, the virtual addresses are converted into real memory addresses.

Why is it needed….

- Before the development of the virtual memory technique, programmers in the 1940s and 1950s had to manage directly two-level storage such as main memory or ram and secondary memory in the form of hard disks or earlier, magnetic drums.

- Enlarge the address space, the set of addresses a program can utilize.
- Virtual memory might contain twice as many addresses as main memory.

Object…

- When a computer is executing many programs at the same time, Virtual memory make the computer to share memory efficiently.
- Eliminate a restriction that a computer works in memory which is small and be limited.
- When many programs is running at the same time, by distributing each suitable memory area to each program, VM protect programs to interfere each other in each memory area.

How does it work…

- To facilitate copying virtual memory into real memory, the operating system divides virtual memory into pages, each of which contains a fixed number of addresses.
- Each page is stored on a disk until it is needed.
- When the page is needed, the operating system copies it from disk to main memory, translating the virtual addresses into real addresses.

Segmentation……

- Segmentation involves the relocation of variable sized segments into the physical address space.
- Generally these segments are contiguous units, and are referred to in programs by their segment number and an offset to the requested data.
- Efficient segmentation relies on programs that are very thoughtfully written for their target system.
- Since segmentation relies on memory that is located in single large blocks, it is very possible that enough free space is available to load a new module, but can not be utilized.
- Segmentation may also suffer from internal fragmentation if segments are not variable-sized, where memory above the segment is not used by the program but is still “reserved” for it.

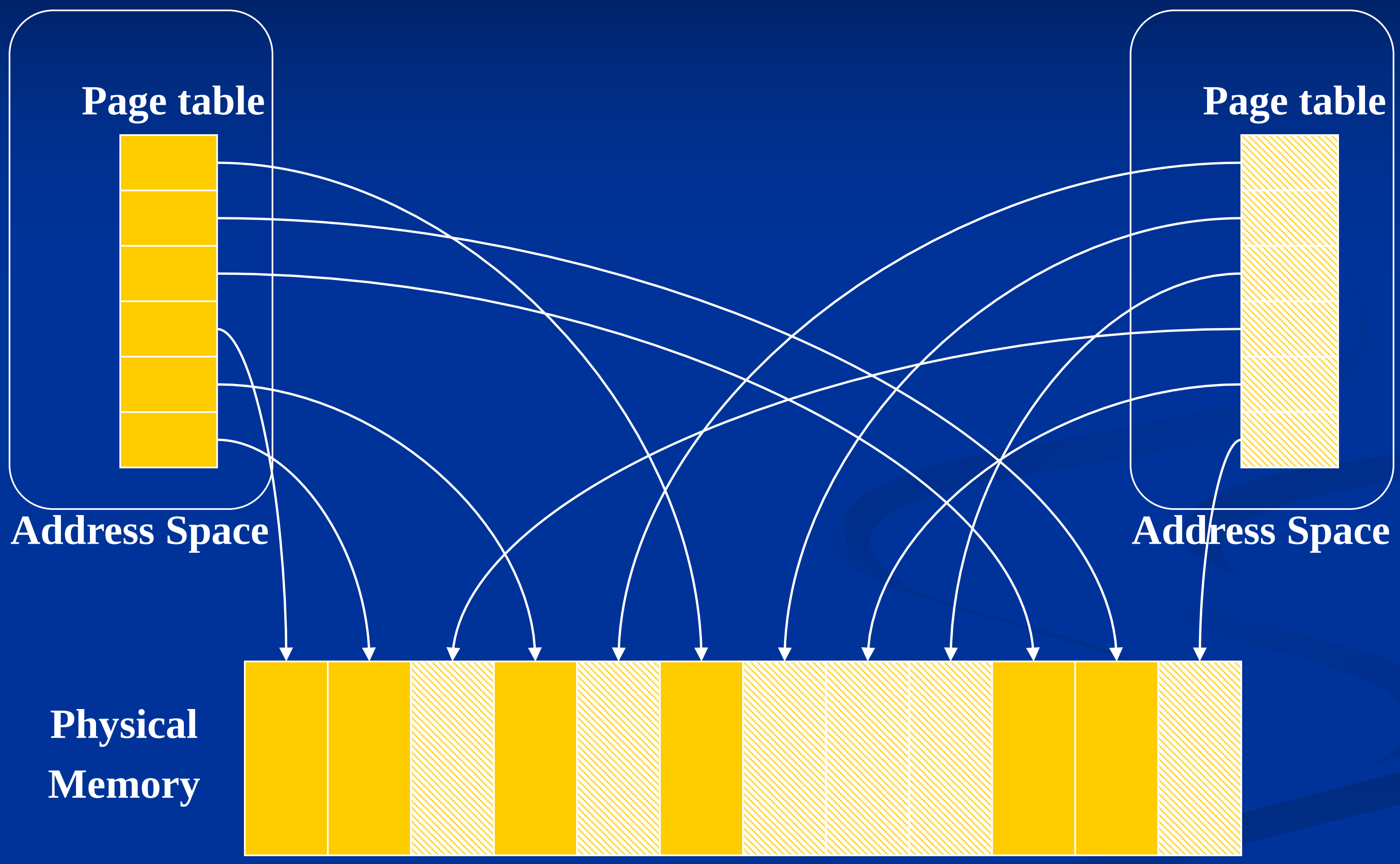
Paging……

- Paging provides a somewhat easier interface for programs, in that its operation tends to be more automatic and thus transparent.
- Each unit of transfer, referred to as a page, is of a fixed size and swapped by the virtual memory manager outside of the program's control.
- Instead of utilizing a segment/offset addressing approach, as seen in segmentation, paging uses a linear sequence of virtual addresses which are mapped to physical memory as necessary.
- Due to this addressing approach, a single program may refer to series of many non-contiguous segments.
- Although some internal fragmentation may still exist due to the fixed size of the pages, the approach virtually eliminates external fragmentation.

Paging……(cont’d)

- A technique used by virtual memory operating systems to help ensure that the data you need is available as quickly as possible.
- The operating system copies a certain number of pages from your storage device to main memory.
- When a program needs a page that is not in maim memory, the operating system copies the required page into memory and copies another page back to the disk.

Virtual Memory (Paging)



Page fault

- An interrupt to the software raised by the hardware when a program accesses a page that is not mapped in physical memory.

- when a program accesses a memory location in its memory and the page corresponding to that memory is not loaded
- when a program accesses a memory location in its memory and the program does not have privileges to access the page corresponding to that memory.

Summary…

- Virtual memory is a common part of most operating systems on computers.
- It has become so common because it provides a big benefit for users at a very low cost.
- benefits of executing a program that is only partially in memory.
- program is no longer constrained by the amount of physical memory.
- ⇒ user would be able to write programs for an extremely large virtual address space.
- more programs could be run at the same time
- ⇒ increase CPU utilization and throughput.
- less I/O would be needed to load or swap each user program
- ⇒ run faster

Logical vs. Physical Address Space

- The concept of a logical *address space* that is bound to a separate *physical address space* is central to proper memory management.
  - *Logical address* – generated by the CPU; also referred to as *virtual address*.
  - *Physical address* – address seen by the memory unit.
- Logical and physical addresses are the same in compile-time and load-time address-binding schemes; logical (virtual) and physical addresses differ in execution-time address-binding scheme.

Memory-Management Unit (MMU)

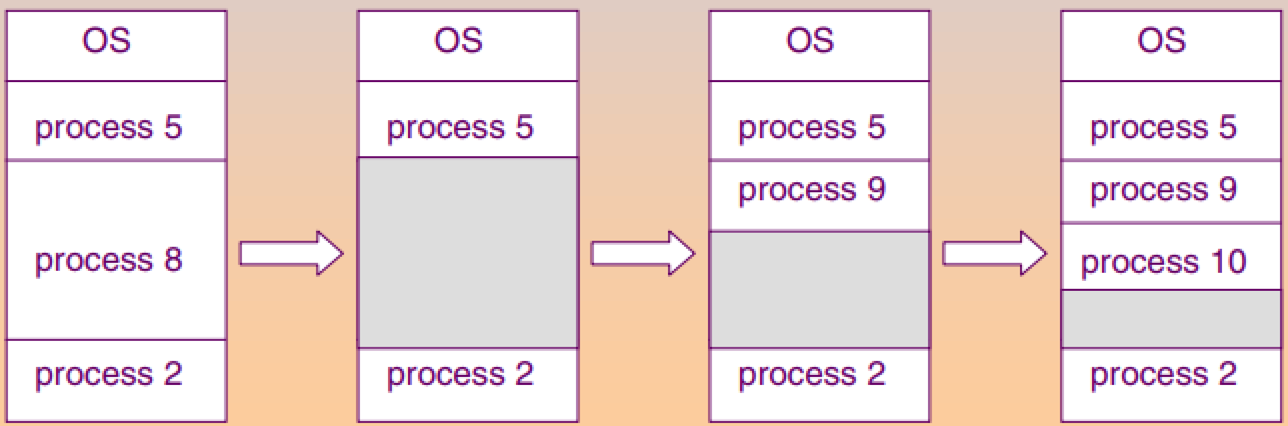
- Hardware device that maps virtual to physical address.
- In MMU scheme, the value in the relocation register is added to every address generated by a user process at the time it is sent to memory.
- The user program deals with *logical* addresses; it never sees the *real* physical addresses.

Contiguous Allocation

- Main memory usually into two partitions:
  - Resident operating system, usually held in low memory with interrupt vector.
  - User processes then held in high memory.
- Single-partition allocation
  - Relocation-register scheme used to protect user processes from each other, and from changing operating-system code and data.
  - Relocation register contains value of smallest physical address; limit register contains range of logical addresses – each logical address must be less than the limit register.

Contiguous Allocation (Cont.)

- Multiple-partition allocation
  - *Hole* – block of available memory; holes of various size are scattered throughout memory.
  - When a process arrives, it is allocated memory from a hole large enough to accommodate it.
  - Operating system maintains information about:
    a) allocated partitions   b) free partitions (hole)



Dynamic Storage-Allocation Problem

How to satisfy a request of size *n* from a list of free holes.

- **First-fit**: Allocate the *first* hole that is big enough.
- **Best-fit**: Allocate the *smallest* hole that is big enough; must search entire list, unless ordered by size. Produces the smallest leftover hole.
- **Worst-fit**: Allocate the *largest* hole; must also search entire list. Produces the largest leftover hole.

First-fit and best-fit better than worst-fit in terms of speed and storage utilization.

Fragmentation

- **External Fragmentation** – total memory space exists to satisfy a request, but it is not contiguous.
- **Internal Fragmentation** – allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used.
- Reduce external fragmentation by compaction
  - Shuffle memory contents to place all free memory together in one large block.
  - Compaction is possible *only* if relocation is dynamic, and is done at execution time.
  - I/O problem
    - ✔ Latch job in memory while it is involved in I/O.
    - ✔ Do I/O only into OS buffers.

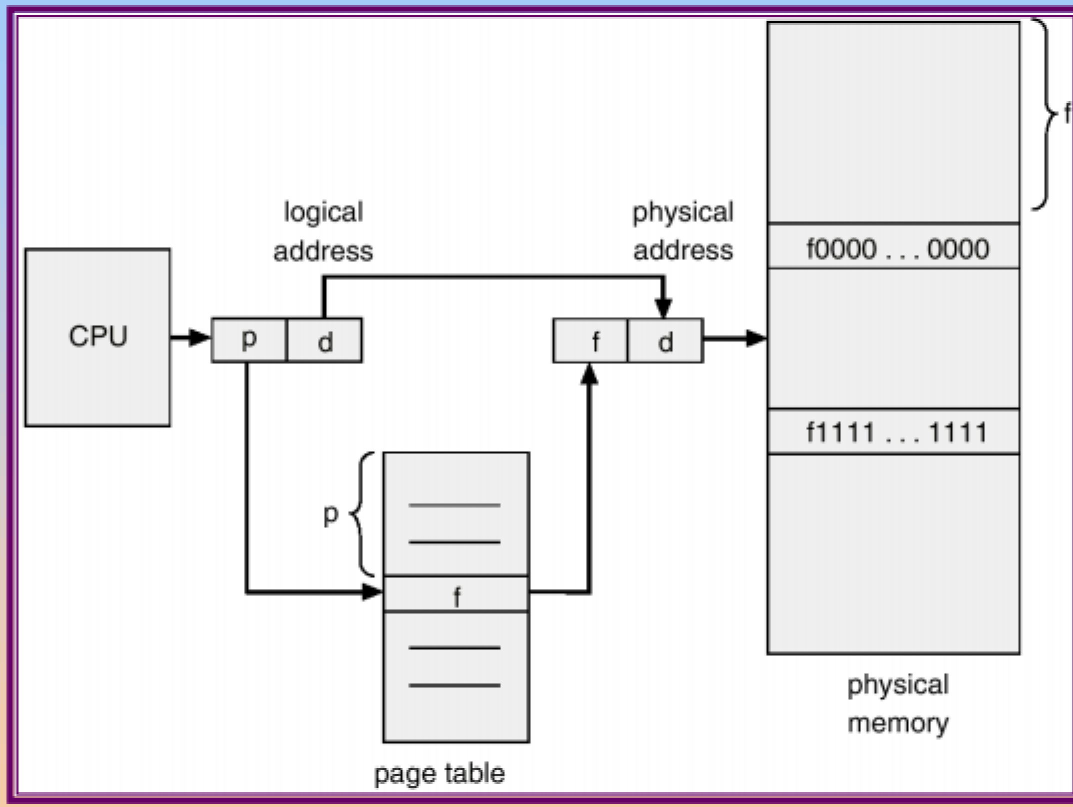
Paging

- Logical address space of a process can be noncontiguous; process is allocated physical memory whenever the latter is available.
- Divide physical memory into fixed-sized blocks called **frames** (size is power of 2, between 512 bytes and 8192 bytes).
- Divide logical memory into blocks of same size called **pages**.
- Keep track of all free frames.
- To run a program of size *n* pages, need to find *n* free frames and load program.
- Set up a page table to translate logical to physical addresses.
- Internal fragmentation.

Address Translation Scheme

- Address generated by CPU is divided into:
  - *Page number (p)* – used as an index into a *page table* which contains base address of each page in physical memory.
  - *Page offset (d)* – combined with base address to define the physical memory address that is sent to the memory unit.

Address Translation Architecture



Paging Example