

Paged Virtual Memory

- ➔ The OS can use disk to simulate larger virtual than physical memory
the pages can be moved between memory and disk (a.k.a paging in/out)

Paging process over time

- Initially, pages are allocated from memory
- When memory fills up, allocating a page requires some other page to be evicted
- Evicted pages go to disk, more precisely to the swap file/backing store
- Done by the OS, and transparent to the application

Improvement : **demand paging** (a.k.a lazy loading) in a page from disk into memory only if an attempt is made to access it (the main memory becomes a cache for disk)

Page Faults

Read/write/execute protection bits : operation not permitted on page

- ➔ The TLB traps to the OS and the OS usually will send fault back up to process, or might be playing games e.g., copy on write, mapped files (coming later in this lecture)

Invalid bits : 2 possible reasons

1. Virtual page not allocated

- ➔ The TLB traps to the OS and the OS sends fault to process (e.g., segmentation fault) TLB traps to the OS (software takes over)

2. Virtual page is allocated in the address space but swapped on disk

- ➔ The TLB traps to the OS and the OS sends allocates frame, reads from disk, maps PTE to physical frame