

Paging

- Page - a chunk of contiguous address from an address space
- Page frame - a chunk of contiguous physical memory
- Typically 4 KB per page

First page - addresses 0 through 4095

Second page - addresses 4096 through 8191

⋮

- Virtual memory mapping is done on a page level

- Page Fault

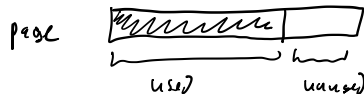
- If a program tries to access a page that is not in main memory (moved to disk), the MMU alerts the OS
- If there are no free pages in RAM, the OS writes a page from RAM to disk and writes the needed page from disk to RAM
- Each page has a page number
 - Assuming a 32-bit virtual address space and a 4kB page size
 - Least significant 12 bits are the offset into the page
 - Most significant 20 bits is the page number

- Page table

- Maps page numbers to page frame numbers
- Must be very fast
- Translating a virtual address to a physical address
 - Replace the page number with the page frame number
 - Offset remains the same

Internal fragmentation

- Processes are unlikely to occupy a multiple of 4096 bytes
- Unused space within a page cannot be used by another process and is wasted



- Smaller page frames
 - Less internal fragmentation
 - Larger page table
- Larger page frames
 - More internal fragmentation
 - Smaller page tables
- 4 KB is a good compromise for modern computers

Segmentation

- A process's memory can be divided into logical segments
- Varies from system to system (OS dependant)
- Common segments
 - Text segment (also called the code segment)
 - Contains the program's instructions
 - Data segment
 - Contains static and global variables
 - Read-only segment
 - Contains constants and literal values
 - Stack segment
 - Contains local variables
 - Heap segment
 - Contains dynamically allocated memory (created with `malloc()` and `friends` in C, created with `new` in C++)

- Unlike pages, segments can be resized
- Segments can have their own address space
- If a process tries to access memory from segments it should not be accessing, the program crashes with a segmentation fault or a bus error