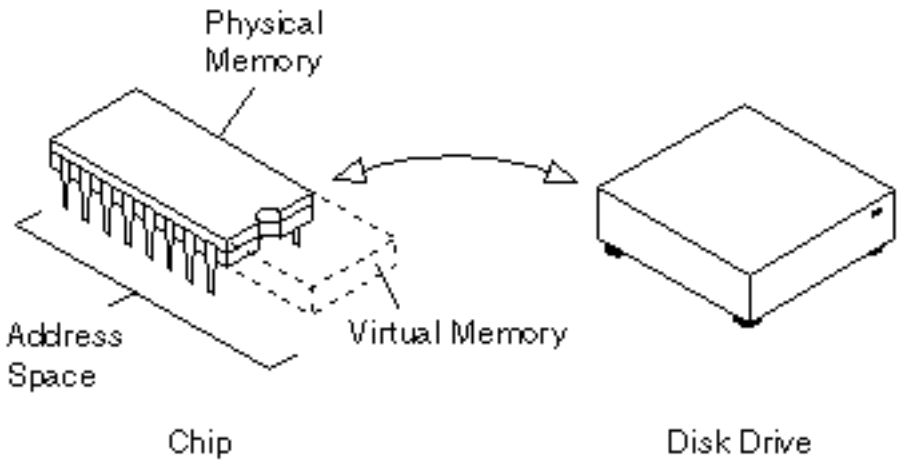
Memory Management

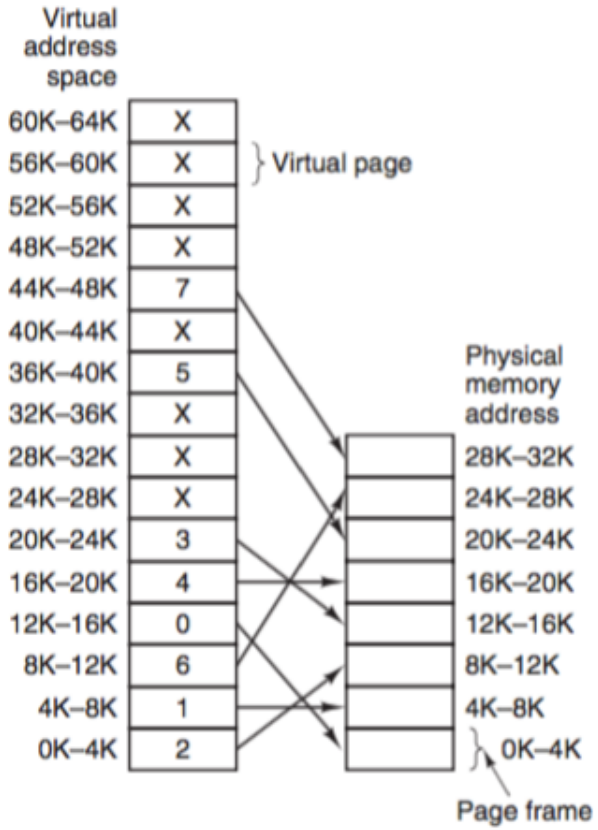
Week 08 – Lab

Virtual Memory



- Processor operates with Virtual Memory addresses
- Actual data (source code + data) is stored in Physical Memory
- Page tables : Virtual Memory -> Physical Memory

Purpose of Virtual Memory



To enlarge address space, the set of memory addresses the system can use

Exercise 1

- Run `*free -t -h*` in the shell or `*vm_stat*` on macOS

- **Mem** represents physical memory size
- **Swap** represents size of memory available for swapping
- **Total** represents virtual memory size

Exercise 1(windows)

- There is a command such as *'free'* but we can get the physical and virtual memory size using the following commands.

  - systeminfo | find "Physical Memory"
  - systeminfo | find "Virtual Memory“

- Note: mobaXterm support **free** command but it has a different implementation compared to the native Linux command.

vmstat / vm_stat

- Reports information about processes, memory, paging, block IO, traps, and cpu activity
- The first report produced gives averages since the last reboot. Additional reports give information on a sampling period of length delay. The process and memory reports are instantaneous in either case

Exercise 2

- Write a C program that runs for 10 seconds. Every second it should:
  - allocate 10 MB of memory
  - fill it with zeros
  - sleep for 1 second
- Compile and run the program in the background (./ex2 &) and run `*vmstat 1*` at the same time. Observe what happens to the memory. Pay attention to **si** and **so** fields.
- *Add comments to your source code with your findings.*
- Hint: use *memset(ptr, value, size)* to fill the allocated memory

top

- Provides an ongoing look at processor activity in real time. It displays a listing of the most CPU-intensive tasks on the system, and can provide an interactive interface for manipulating processes

Exercise 3

- Run `*top -d 1*` or `*top -i 1*` *on macOS*
- Run ex2 program in the background and then run `*top*`
- *Add comments to your source code with your findings.*

- Note: for windows users, can run the command ***tasklist****, ex: 'tasklist /v /FI "STATUS eq running"'*

getrusage()

- C function from <sys/resource.h> library to monitor application's memory usage. Refer to `man 2 getrusage`

```
int getrusage(int who, struct rusage *usage);
```

Exercise 4

- Write a C program that runs for 10 seconds. Every second it should:
  - allocate 10 MB of memory
  - fill it with zeros
  - print memory usage with getrusage() function
  - sleep for 1 second

Exercise 5

- What is the difference between a physical and a virtual address? Describe using **your own words**. Save your answer to ex5.txt

Exercise 6

- A machine has 16-bit virtual addresses. Pages are 8 KB. How many entries are needed for a single-level linear page table? Explain your computations. Save your answer to ex6.txt

(Hint: Modern Operating Systems, 3.3.2)

Extra exercise

- Download and run Memory Management Simulator
- Installation instructions:
  http://www.ontko.com/moss/memory/install_unix.html
- Download:
  http://www.ontko.com/moss/memory/memory.tgz
- User guide:
  http://www.ontko.com/moss/memory/user_guide.html

Extra exercise

- Modify commands file so that the last instruction would write to the 32-nd virtual page in memory. Notice the swapping of virtual page to a physical memory