# Virtual Memory Revisited

Lecture 20
Class 22 of 28 | April 4th 2023 | COMP 211-002 | Joshua Bakita

# Welcome!

Today:

- Examining `mmap()` and `fread()`
- Page faults, syscalls, and virtual memory
- The `make` command

Logistics:

- Post-midterm review guide
- A3 extension
- A5 changes & extension

Fun fact...

*The NC State Constitution states, "...the benefits of The University of North Carolina... as far as practicable, be extended to the people of the State free of expense."*

*And NC spends more per-capita on higher education than almost any other state.*

# Virtual Memory

What is it useful for, beyond `mmap()`?

## Virtual Memory

# What and why?

Every address your program sees, what's stored in every pointer, is a *virtual address*.

The operating system controls what *physical address*, if any, every virtual address corresponds to.

- ➔ Memory-mapped files
  - ◆ mmap()
- ➔ Memory protection
  - ◆ One process cannot access another another's memory
- ➔ Shared memory
  - ◆ Why keep multiple copies of a file around in-memory?
- ➔ Memory oversubscription
  - ◆ What if we need more memory than our system has available?

# A page fault!

What happens if you access a virtual address with no currently mapped corresponding physical address?

# System Calls (syscalls)

System Calls

# When do they occur?

Whenever you need to perform a privileged operation

- ➔ Like accessing a storage device
- ➔ Signaling another process
- ➔ Updating virtual memory mappings
- ➔ Exiting your program
- ➔ ...

When you make a syscall as part of the library functions `fopen()`, `fread()`, `exit()` or similar, you run operating system code, which may suspend your application as part of performing the operation requested.

Trace for cat with fread():
https://www.cs.unc.edu/~jbakita/teach/comp211-s23/l20/kutrace_fread.html

**Some syscall and page fault examples…**

# Via KUtrace

Trace for cat with mmap():
https://www.cs.unc.edu/~jbakita/teach/comp211-s23/l20/kutrace_mmap.html

# Questions?

Contact:
Email: hacker@unc.edu
Twitter: @JJBakita
Web: https://cs.unc.edu/~jbakita

Old Well, University of North Carolina at Chapel Hill, Winter 2017