Debugging and Profiling

Lecture 13
Feb 28th 2023 | COMP 211-002 | Joshua Bakita

Welcome!

Today:

- More on I/O
  - `mmap()`
  - Performance profiling
- Debugging review

Logistics:

- 54% of the class has started on Assignment 3
- The bottom 6% of the class got an email warning last week—please do come meet with me before Friday

Fun fact...

Most terminals support `Ctrl+w` to delete the last word you typed, and `Ctrl+u` to delete the whole line.

File I/O… Differently

Is there something less clunky than `fread()`?

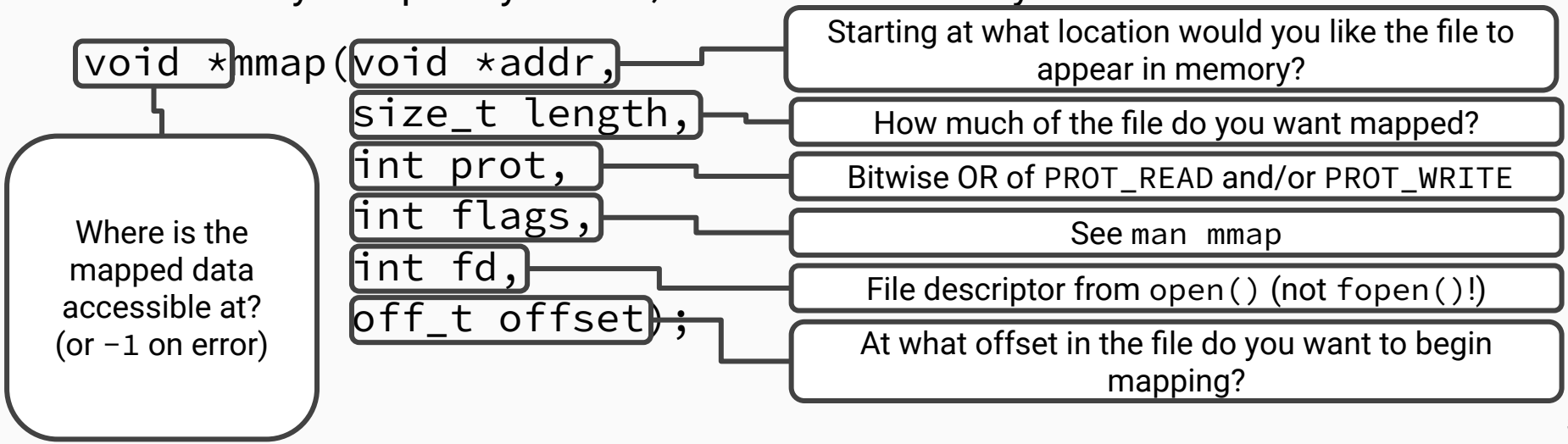
File I/O… Differently

Demo: Revisiting `cat`

File I/O... Differently

Memory Mapping

Configure memory such that you can directly read/write to a file, as though it were already completely read in, and is automatically written out.



File I/O… Differently

Writing `cat` with `mmap()`

Performance Profiling

A key aspect in architectural decisions!

Performance Profiling

`time` and `perf`

Debugging Revisited

Likely relevant to Assignment 3!

Command Line

`valgrind prog`	Run `prog` with valgrind
`gdb prog`	Start the GNU Debugger on `prog`
`info thing`	View detailed manual for `thing`
`xxd file`	Print `file` as hexadecimal
`wget addr`	Download file from `addr`
`rm file`	Delete `file`
`cd dir`	Move to `dir`
`cat file`	Print contents of `file`
`cp fileA fileB`	Copy `fileA` to `fileB`

Vim Commands (Normal Mode)

`dd`	Delete current line
`D`	Delete from cursor to end-of-line
`>>`	Increase indent
`<<`	Decrease indent
`O`	Add line above cursor and enter insert mode
`o`	Add line below cursor and enter insert mode

For Your ~/.vimrc Config File

```
set cindent
set nowrap
```

Key GNU Debugger (GDB) Commands—From L7

Full command name

Shorthand

Access the full GDB manual via `info gdb` on the command line

Control Flow

`backtrace`	`bt`	List all stack frames
`select <frame#>`	`sel`	Select a stack frame as your context
`next`	`n`	Execute the next line from your context
`step`	`s`	Execute one line
`list`	`l`	Print source code

Data

`print <expr>`	`p`	Execute expression and print result (can modify data)
`info locals`	`i lo`	Print value of every local variable in your stack frame
`x <addr>`	`x`	Print bytes at `addr` in memory
`whatis <expr>`	`wha`	Print type of `expr`

Breakpoints

`break <file>:<l>`
`break <function>`
`break <function> if <condition>`
	`b`	Set a breakpoint with optional condition at a location or function

`info breakpoints`	`i b`	List all breakpoints
`delete <breakpoint number>`	`d`	Delete a breakpoint
`continue`	`c`	Resume execution

Admin

`run <args>`	`r`	Run local program
`quit`	`q`	Exit GDB
`help <cmd>`	`h`	Print quick reference for a command
`set history save`		Save command history

How Does I/O Really Work?

A sampling from one of my research presentations…

Questions?

See office hour calendar on the website for availability.

Assignment 3 due Tuesday!

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

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