IPC and Processes

Lecture 24
Class 26 of 28 | April 20th 2023 | COMP 211-002 | Joshua Bakita

Front Matter

Welcome!

Today:

- ➔ Logistics
- ➔ Inter-Process Communication
- ➔ Assignment 5 Functions

Logistics:

- ➔ For regrade rqs., prefer Gradescope or Pizza
- ➔ Research opportunity if you get an A/A-

Fun fact…

You can run any shell command from inside vim via the ! command in normal mode.

Example:
`:!gcc modify.c -o modify`
to compile `modify`

Logistics

Final Exam:

- ➔ Tues, May 2nd, Noon to 3 PM
- ➔ Comprehensive, emphasis on first half of the course
- ➔ Format like midterms, drawing from class, readings, and assignments
- ➔ 20% Weight
  - ◆ Min 40% for non-F overall grade
- ➔ Can replace lower of two midterms
- ➔ ARS Extended Time in SN314

Assignments:

8% each

A3. Final grades very soon

A4. Late due date now passed.

- ○ Target: Grades next Thursday

A5. Due by the final exam — 10%+

A6. Cut

- ○ Previously extra-credit only
- ○ Lots of extra credit opportunities in A5
- ○ Come talk with me about GPUs!

If you have a final conflict, or 3 in 24h exception, form: https://eef.oasis.unc.edu/

Why care about edge cases?

Why care about edge cases?

When edge cases are not handled…

Some space missions that failed due to a software bug:

- Phobos 1 – Test code left enabled
- Ariane-5 – Integer overflow
- Mars Climate Orbiter – Navigation failure due to treating imperial units as metric
- Mars Global Surveyor – Memory corruption

The results can be disastrous! Get in the habit of doing it right.

Don't forget to check your code for memory errors with valgrind!

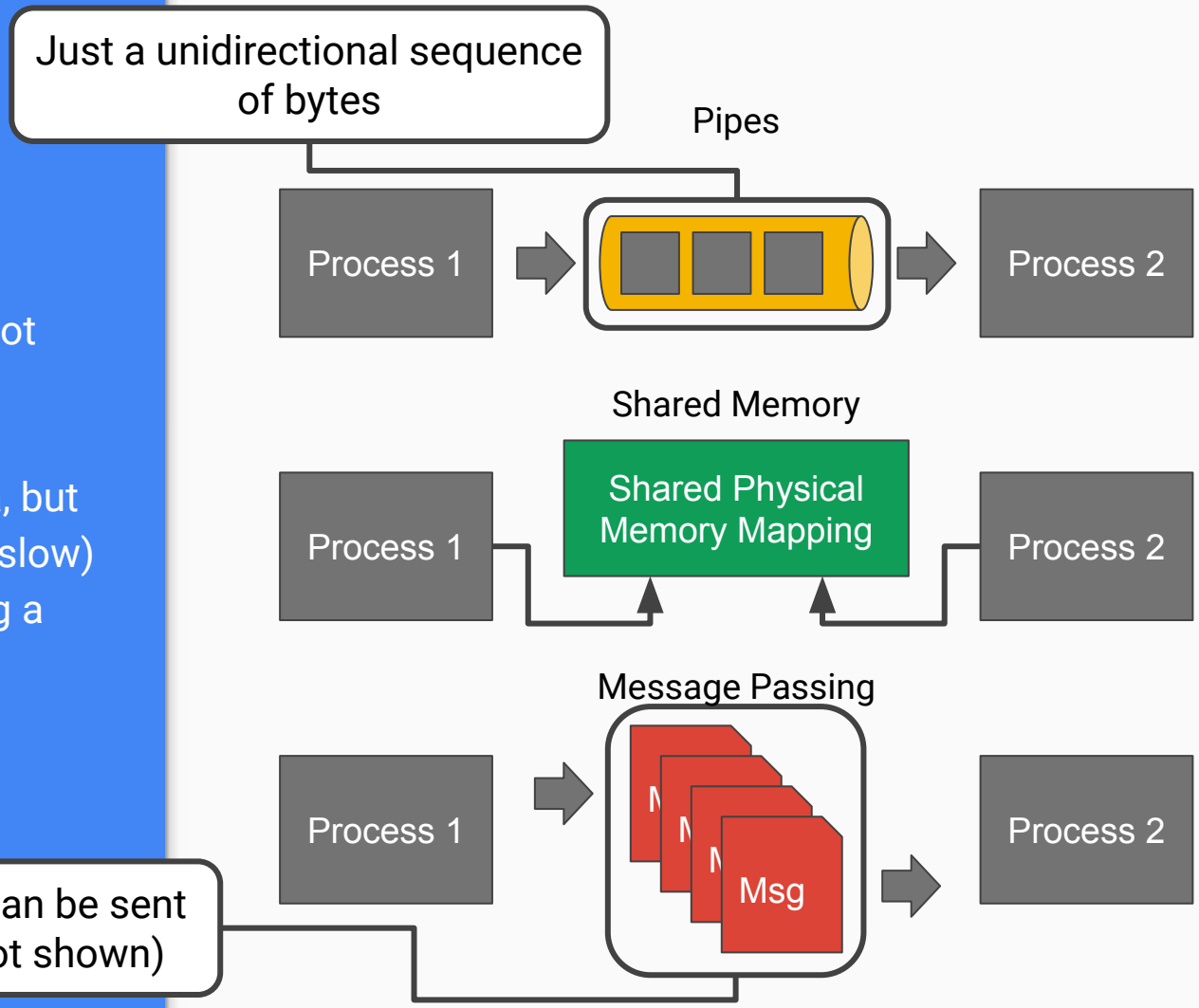
Inter-Process Communication (IPC)

Beyond Signals

Inter-Process Communication

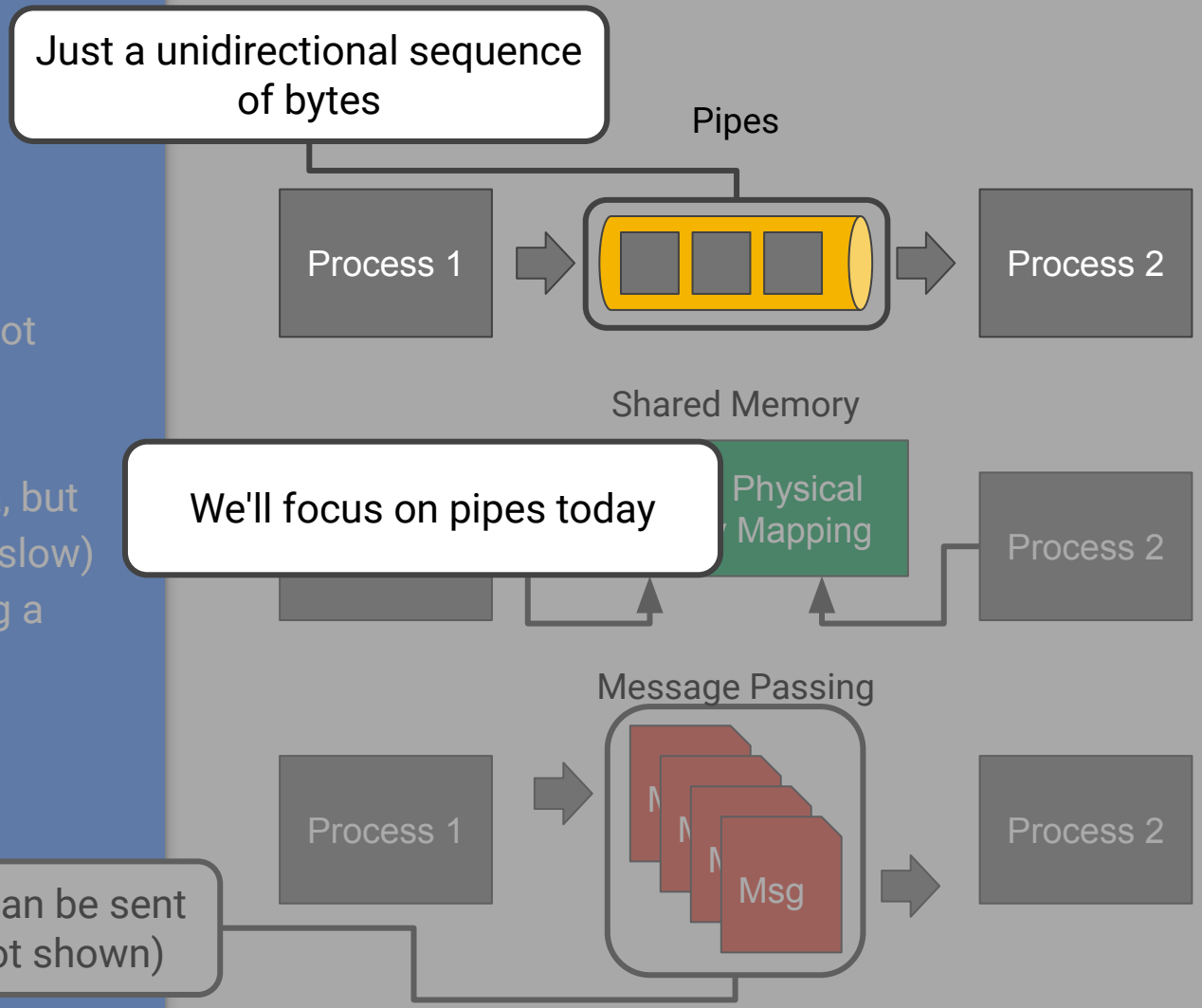
What and why?

- ➔ Signals are often not enough
- ➔ What if we want to communicate data, but want to avoid the (slow) process of creating a file on disk?

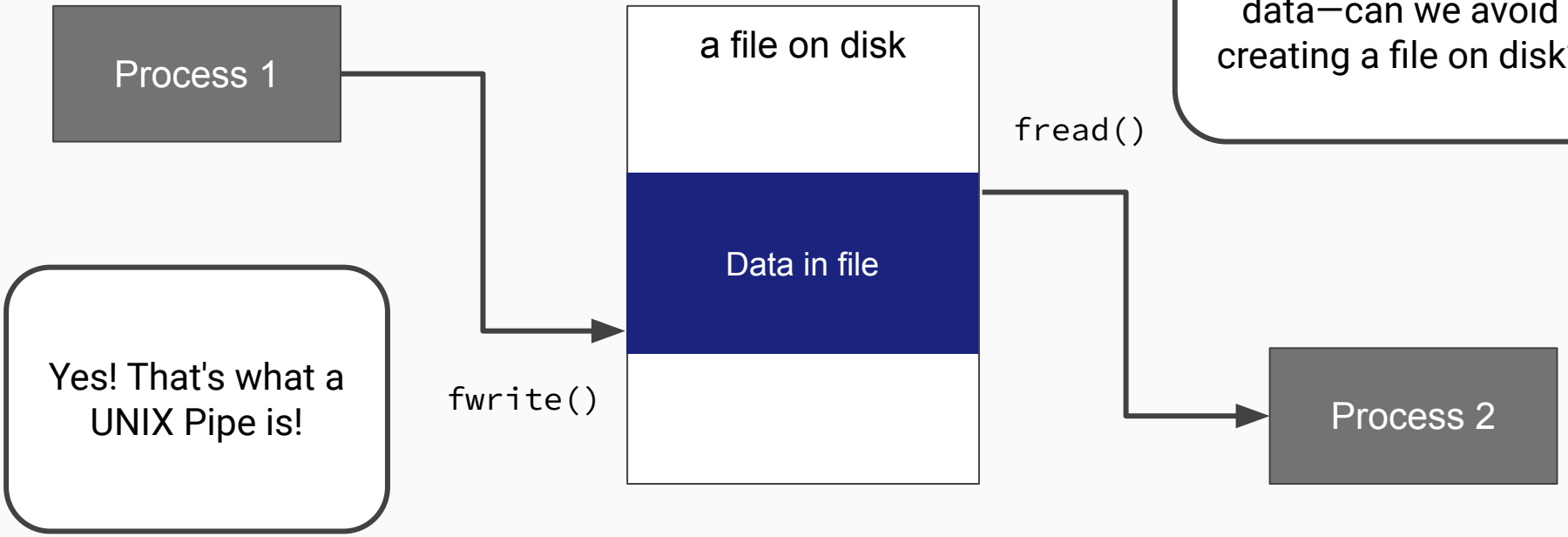


What and why?

- ➔ Signals are often not enough
- ➔ What if we want to communicate data, but want to avoid the (slow) process of creating a file on disk?



How do we get to pipes?



You've already been using pipes

`stdin`, `stdout`, and `stderr` are (by default) all just open pipes, connected to or from your terminal!

UNIX Pipes API

- Create pipes via the `pipe()` function
- This gives to file descriptors (FDs):
  - One that can be written to
  - One that can be read from
- Simply `read()` or `write()` using the respective FD
  - Or use `fdopen()` to convert it to a `FILE*` (`man fdopen`)
- See `man 2 pipe` and the readings on the course webpage

Inter-Process Communication

Demo

Let's also look at a few things for A5, like `fork()`, `execvp()`, `wait()` and maybe `dup2()`

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