Advanced Programming in the UNIX Environment

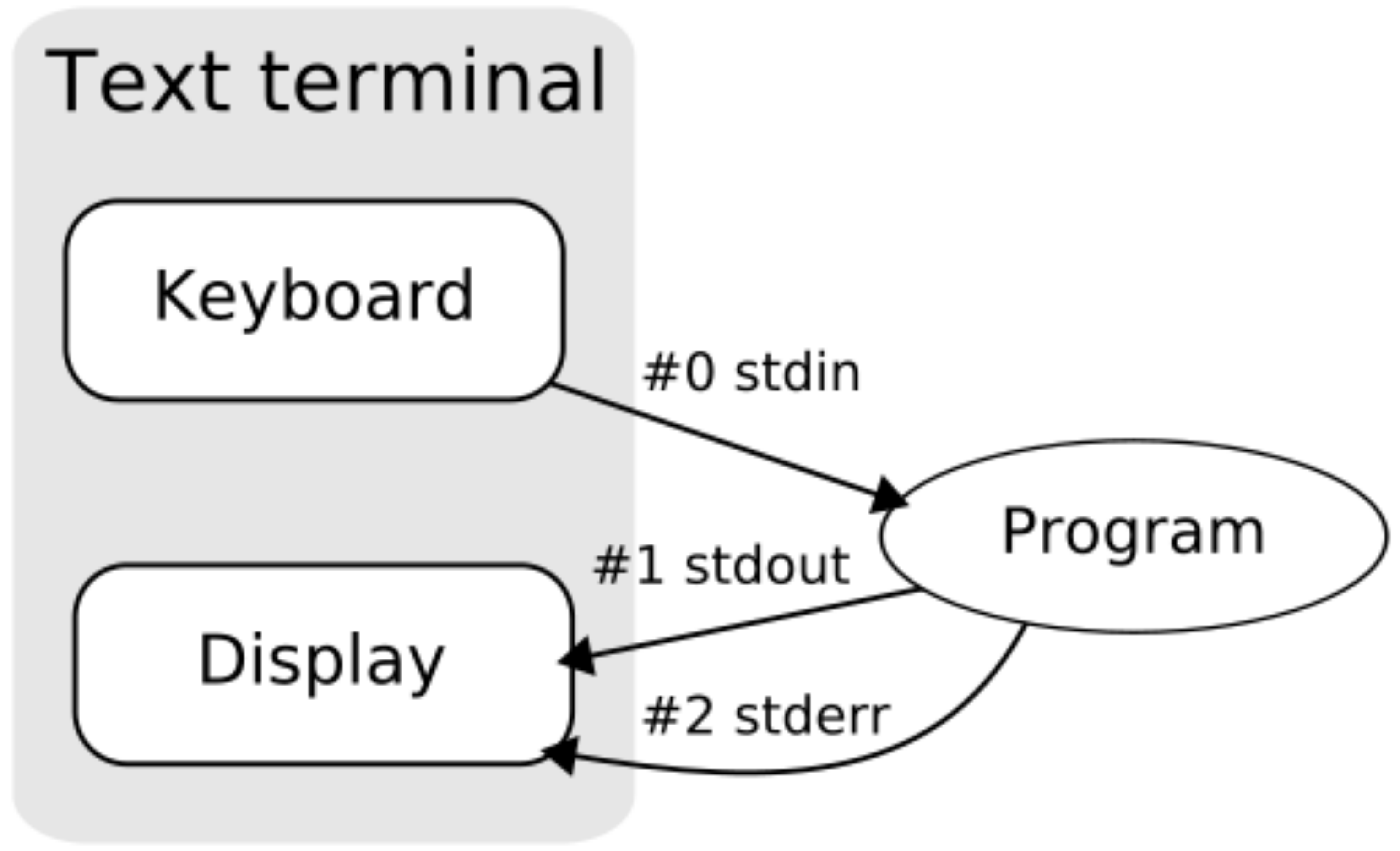
Week 02, Segment 1: File Descriptors

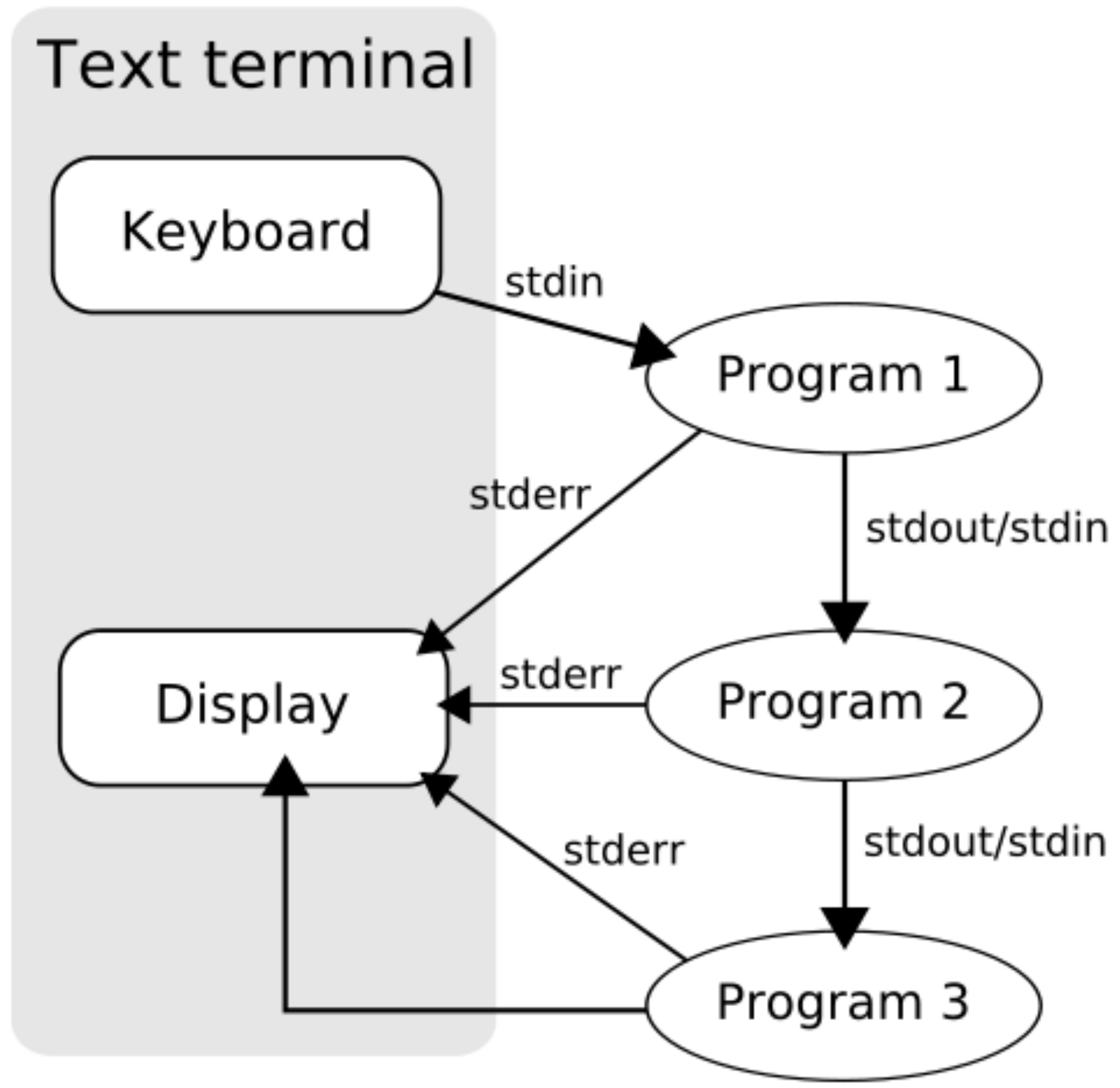
Department of Computer Science
Stevens Institute of Technology

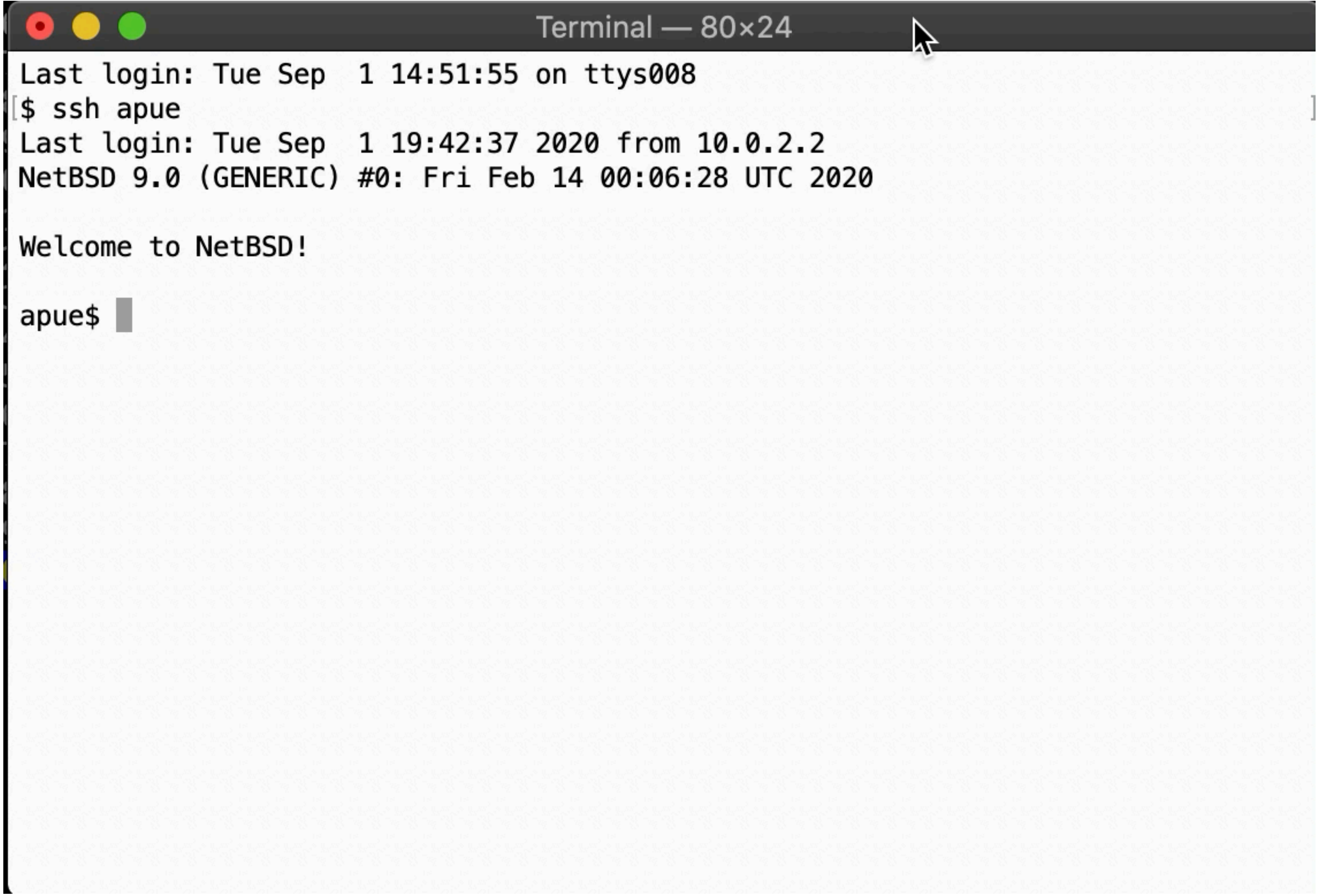
Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

File Descriptors - see `stdio(3)`

- A *file descriptor* (or file handle) is a small, non-negative integer which identifies a file to the kernel.
- Traditionally, `stdin`, `stdout` and `stderr` are `0`, `1` and `2`, respectively, but relying on magic numbers is bad practice. Use STDIN_FILENO, STDOUT_FILENO, and STDERR_FILENO instead.







```
Last login: Tue Sep  1 14:51:55 on ttys008
$ ssh apue
Last login: Tue Sep  1 19:42:37 2020 from 10.0.2.2
NetBSD 9.0 (GENERIC) #0: Fri Feb 14 00:06:28 UTC 2020

Welcome to NetBSD!

apue$ 
```

Lessons

You can spend a surprising amount of time on a simple question like "How many file descriptors can a process have open?"

- You can't always rely on values being defined for you.
- The defined value may not actually apply to your process.
- Constants required by the standard may, while present, not actually be useful.
- Use `sysconf(3)` / `getrlimit(2)` for runtime values, but keep in mind that the result may change from invocation to the next.
- Get in the habit of writing code to verify / check your understanding.
- Testing across Unix versions can help illustrate difference.

Exercises

Run https://stevens.netmeister.org/631/openmax.c on different Unix versions.

What happens if you run 'ulimit -n 0'? Why?

If, as root, you set 'ulimit -n unlimited', what number will be used? Why?

What, if anything, does any of this have to do with _POSIX_OPEN_MAX?

See also: https://stevens.netmeister.org/631/fd-exercise.html

From here on: Standard I/O: open(2) and close(2)