Wrap up

Lecture 26
Class 28 of 28 | April 27th 2023 | COMP 211-002 | Joshua Bakita

Front Matter

Welcome!

Today:

- ➔ Notes on pipe()
- ➔ Assignment 4 Style
- ➔ Conclusion

Logistics:

- ➔ A3 Auto. & A4 Style Up
- ➔ Research interest? See form on website
- ➔ Get <70% overall on an assignment? Fix it and fill out the form on the website to get back up to 30% of what you lost

Fun fact...

Alex G. is currently the #1 leader on the Assignment 5 hacked save leaderboard.

Last Time

```c
#include <errno.h>
#include <error.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/wait.h>
#include <string.h>

#define PROG_TO_RUN "./rank"

int main(int agrc, char** argv, char** envp) {
    pid_t pid = fork();
    int my_pipes[2];
    pipe(my_pipes);
    if (pid) {
        printf("in parent\n");
        int res;
        close(my_pipes[0]);
        char * files = "tetris_quicksave.bin\n";
        write(my_pipes[1], files, strlen(files));
        wait(&res);
    } else {
        printf("in child\n");
        close(my_pipes[1]);
        dup2(my_pipes[0], 0);
        char* my_args[2]
        my_args[0] = PROG_TO_RUN;
        my_args[1] = "score";
        my_args[2] = "3";
        my_args[3] = NULL;
        execvp(PROG_TO_RUN, my_args);
        error(1, errno, "%s failed to run", PROG_TO_RUN);
    }
    retu
}
```

Corrected

```c
#include <errno.h>
#include <error.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/wait.h>
#include <string.h>

#define PROG_TO_RUN "./rank"

int main(int agrc, char** argv, char** envp) {
    int my_pipes[2];
    pipe(my_pipes);
    pid_t pid = fork();
    if (pid) {
        printf("in parent\n");
        int res;
        close(my_pipes[0]);
        char * files = "tetris_quicksave.bin\n";
        write(my_pipes[1], files, strlen(files));
        close(my_pipes[1]);
        wait(&res);
    } else {
        printf("in child\n");
        close(my_pipes[1]);
        dup2(my_pipes[0], 0);
        char* my_args[4];
        my_args[0] = PROG_TO_RUN;
        my_args[1] = "score";
        my_args[2] = "3";
        my_args[3] = NULL;
        execvp(PROG_TO_RUN, my_args);
        error(1, errno, "%s failed to run", PROG_TO_RUN);
```

Corrected code:
https://cs.unc.edu/~jbakita/teach/comp211-s23/l25/class_demo.c

Assignment 4 Style

Great job!

First-pass style grades posted

Much improved!

Some use "E*X*" comment notation. The number in place of *X* denotes the respective comment.

Comment codes:

E-1. Repeating declarations from headers
E0. Don't include commented-out code
(see class 25 slides 4, 5, and 6 for E1-E18)
E19. Use of malloc for small variables
E20. Usage messages are best practice if the wrong number of arguments is specified

Closing Comments

What we learned

- How to write C programs
- How to survive on the command line
- What pointers are, and how to use them
- Basics of GDB and valgrind for debugging
- How data is stored and represented
- Complex data structures, like union and struct
- What compiling does
- How to read and write files

What we learned

What 2s complement is, and how to use it

Basics of linking

Static, heap, and static memory

Costs of syscalls

The what and why of virtual memory

Use and downsides of file systems

`mmap()`

How storage devices execute I/O

What we learned

The dangers of dependent loads

The importance of locality

Speed of information in a computer

Need for caches

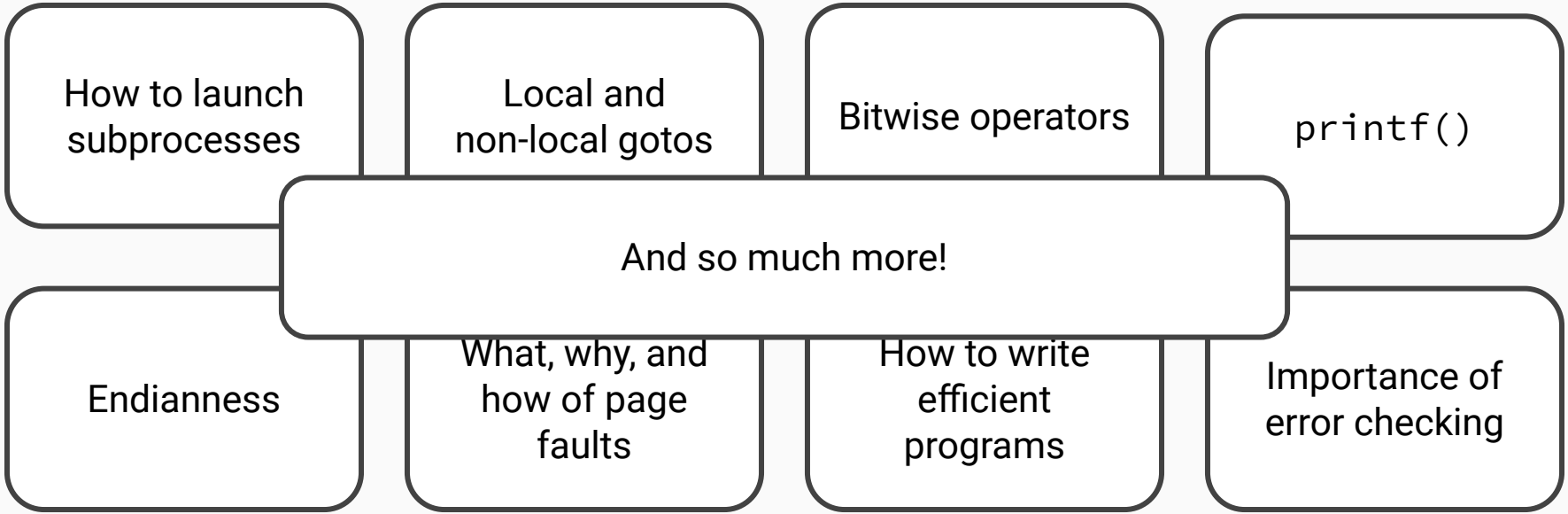
What the major components of a computer are

How to use `make`

How to do IPC in a computer?

Dangers of memory corruption

What we learned



Takeaway:

You learned *far* more than is typically taught in COMP 211

Great systems programming classes to consider:

- Operating Systems
- 2D Graphics
- Compilers
- Digital Logic & Computer Design

Thank you all!

See many of you at the final review session at 1:30 PM on Saturday in SN014.

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

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