Introduction
Agenda

• Introductions
• Reading
• Course information
• C practice
• Assignment
Introduce yourself by completing these statements

My name is

I use the pronouns (they/them/theirs, she/her/hers, he/him/his)

My favorite C operator is
Reading
Questions

• What is an operating system?

• What duty does operating system performs?
Responses

• What is an operating system?
  – A program (body of software) that provides services to user programs

• What duty does operating system performs?
  – Manage resources (e.g., CPU, memory, disk)
  – Make system easy to use
  – Protect programs from each other
  – Schedule programs
  – Detect errors and handle them
Operating System

• Provide a standard library to user programs
  – The library contains system calls
  – Allocate memory, access a file

• OS is resource manager
  – Many programs run (share CPU)
  – Many programs access their own data and instructions (share memory)
  – Many programs access devices (share disk)
Questions

• What are the three aspects of OS that our textbook discusses about? Give an example for each
Responses

• What are the three aspects of OS that our textbook discusses about? Give an example for each
  – Virtualization: OS gives a program an illusion of owning a resource
  – Concurrency: OS runs multiple programs at the same time
  – Persistence: OS stores data permanently on a hard drive
CPU Virtualization

/home/hajiamini/courses/CSC213/ostep-code/cpu.c

gcc -Wall cpu.c -o cpu
(./cpu “A” &); (./cpu “B” &); (./cpu “C”)

Terminate programs:
ps aux | grep cpu
kill <program_number>

• OS provides an illusion that system has many CPUs.
• Each program thinks that it owns CPU.
Concurrency

Find threads.c in /ostep-code directory

gcc -Wall threads.c -o threads -pthread
./threads 800000

- Pthread_create
  - Creates two threads
  - Threads are *worker* functions.

- *The counter* variable is incremented by both threads.

- *counter* is incremented with 3 steps:
  - Load
  - Increment
  - Write

- These steps may not complete at once.
Persistence

• When system crashes, everything in memory is lost.

• Store data persistently
  – Hardware (input/output device): hard drive
  – Software (OS): file system
    • Example system calls: open( ), write( ), close( )
OS Goals

• Build up abstraction
  – Provide system calls to make systems easy to use

• Minimize overhead
  – Avoid extra time and space

• Provide isolation
  – A program does not affect the behavior of other programs.
Course Information
General Information

- **Course website**
  
  www.cs.grinnell.edu/~hajiamini/courses/CSC213

- **Textbook**
  
  Operating Systems: Three Easy Pieces
  by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci Dusseau

- **Class mentors**
  
  Shuyi Qi, Pouya Gholami (Section 1)
  Ben Gafford, Zander Otavka (Section 2)
General Information

• Preparation
  – Read assigned chapters before coming to class
  – Look at Schedule

• Attendance
  – Three free absences; 1.5% penalty for each additional absence

• Resources for getting help
  – Office hours: Tuesday and Thursday, 3-5pm
  – Mentor sessions: 2 sessions per week
General Information

• Labs
  – Every Friday
  – Work with a partner

• Assignments
  – Weekly
  – Work individually

• See due dates in Schedule
General Information

• Exams
  – Quizzes (weekly)
  – Midterm exam (before spring break)
  – Final exam
    ▪ May 13, 2-5pm (Section 1)
    ▪ May 14, 9am-noon (Section 2)
C Practice
Question 1

What is the output of the program?

```c
#include <stdio.h>
int main( ) {
    for( int i = 1; i < 5; i++ ) {
        printf( "Hello world!\n" );
    }
    return 0;
}
```
Question 2

What is the output of the program?

```c
#include <stdio.h>
int main( ) {
    for( unsigned int i = 10; i >= 0; i-- ) {
        printf( “Hello world!\n” );
    }
    return 0;
}
```
Question 3

What is the output of the program?

```c
#include <stdio.h>
int main( ) {
    int x = 1;
    int y = 2;
    int *z = &x;
    y = *z + 2;
    z = &y;
    *z = x;
    printf( "%d %d %d\n", x, y, *z );
    return 0;
}
```
Question 4

What is the output of the program?

```c
int main( ) {
    int *p = malloc( sizeof( int ) );
    int *q = malloc( 2*sizeof( int ) );
    int *r = &q[1];
    *p = 9;
    *q = 11;
    *r = 4;
    printf( "%d %d %d\n", *p, q[0], q[1] );
    return 0;
}
```
Question 5

What is the output of the program?

```c
int main( ) {
    int *arr = malloc( 10*sizeof( int ) );
    for( int i=0; i<10; i++ ) {
        arr[i] += i;
    }

    for(int i=0; i<10; i++){
        printf( "%d %d\n", i, arr[i] );
    }
    return 0;
}
```
Welcome Assignment