

# CompSci 2SD3

## Tutorial #2

**TA: Jatin Chowdhary**

**DATE: February 1st, 2022**

# Announcements (1)

- Starting February 7th, 2022, we will be going 100% in-person for lectures/labs/tutorials/etc.
  - However, *everything*\* will be recorded so don't feel pressured to come in-person (and risk yourself plus the people around you)  
*\*[(Every tutorial run by me will be recorded)]*
- Assignment #1 is due next week; February 9th, 2022
  - Including today, you have 8 days
    - **However, you (my students) should finish by Friday**
- The other TAs have code in their slides
  - Avoid using this unless you are stuck
    - *It is a great resource, but don't lean on it too much*

# Announcements (2)

- The other TAs have code in their slides
  - Avoid using this unless you are stuck
    - Some of you are probably already aware of this
- Do the assignment legitimately
  - Get good at **C**
    - **C** is the greatest programming language
  - Unlike your other courses, the stuff you will learn here is relevant
- Don't even take a “quick” peak at the other slides
  - There is no such thing as “I’m just gonna eat one (potato) chip”
    - If you open the bag, you’re gonna finish the (entire) bag

**Any Questions?**

# Itinerary

- Announcements
- Assignment #1 Review/Walkthrough
  - Where to find it?
  - How to get it via *2SD3fetch*?
  - What to do?
  - How to submit via *2SD3submit*?
  - Corrections and clarifications
  - Questions
  - Correlation description with *TODO*

# Getting Started

- Everything is on the [course website](#)
  1. Login with your student number and password
  2. Click on “*Assignments+Tests*”
  3. Select “*Assignment 1*”
- Basic rules:
  - Using code you found on the Internet (*i.e. Stackoverflow*) is fine, but make sure you cite it
    - *i.e. “I got this snippet of code from abc-tutorials.com”*
  - Avoid discussing/reviewing the assignment with each other
    - Try to be legit – being able to program gets you into *FAANG*

# Other Stuff

- If you're unable to submit after the deadline (*i.e. Emergency, Issues, etc.*), send an email to Dr. Franek ([franek@mcmaster.ca](mailto:franek@mcmaster.ca)), with all the files related to the submission.
  - Include relevant information like:
    - Reason (*i.e. Emergency, Issues, Etc.*)
    - Course code
    - Full name
    - Student number

# Fetch The Assignment

- Create a fetch script
  - Instructions are on the website
    - But I'll go over it for fun
      - *i.e. 2SD3fetch*
- Note:
  - To execute the script, after giving it write permissions (*i.e. `chmod u+x`*), type the following: **`./2sd3fetch`**
    - Side note: Type **`chmod u+x 2sd3fetch`** to make the file an executable
  - You must fetch one file at a time



# The Files

- After fetching, you should have the following:
  - *log.h*
  - *log.c*
  - *job.c*
  - *assgn1\_skel.c*
  - *makefile*
- Note:
  - The only file you need to modify is **assgn1\_skel.c**
    - You can modify the makefile (*i.e. Add “make clean”*)
    - Do not modify the other files; reading them is fine
  - Make sure you rename **assgn1\_skel.c** to **assgn1.c**
    - If you forget to do this, the makefile won't work

# Submission (1)

- Let's make a quick edit (*i.e. Print something*) and submit the assignment
  - Add **printf("Submission test");** to the top of the *main()* function
    - Not literally on top, but the first line after: *main(...)* {
- Note:
  - Always compile your program to make sure it works
    - Broken/buggy code will be heavily penalized

# Submission (2)

- Create the submission script
  - Instructions are on the course website
    - But I'll go over it
      - *i.e. 2SD3submit*
- The only file you need to submit is:
  - **assgn1.c** // This is the file you make amendments to; the C source file
- Note:
  - Don't forget to add `./` in front of *2SD3submit*
    - Similar to *2SD3fetch*
  - Always verify your submission on the course website
    - Login to the course website and double check that your latest submission is there
      - View the contents of the file to make sure you submitted the correct file
    - Do not solely rely on the script

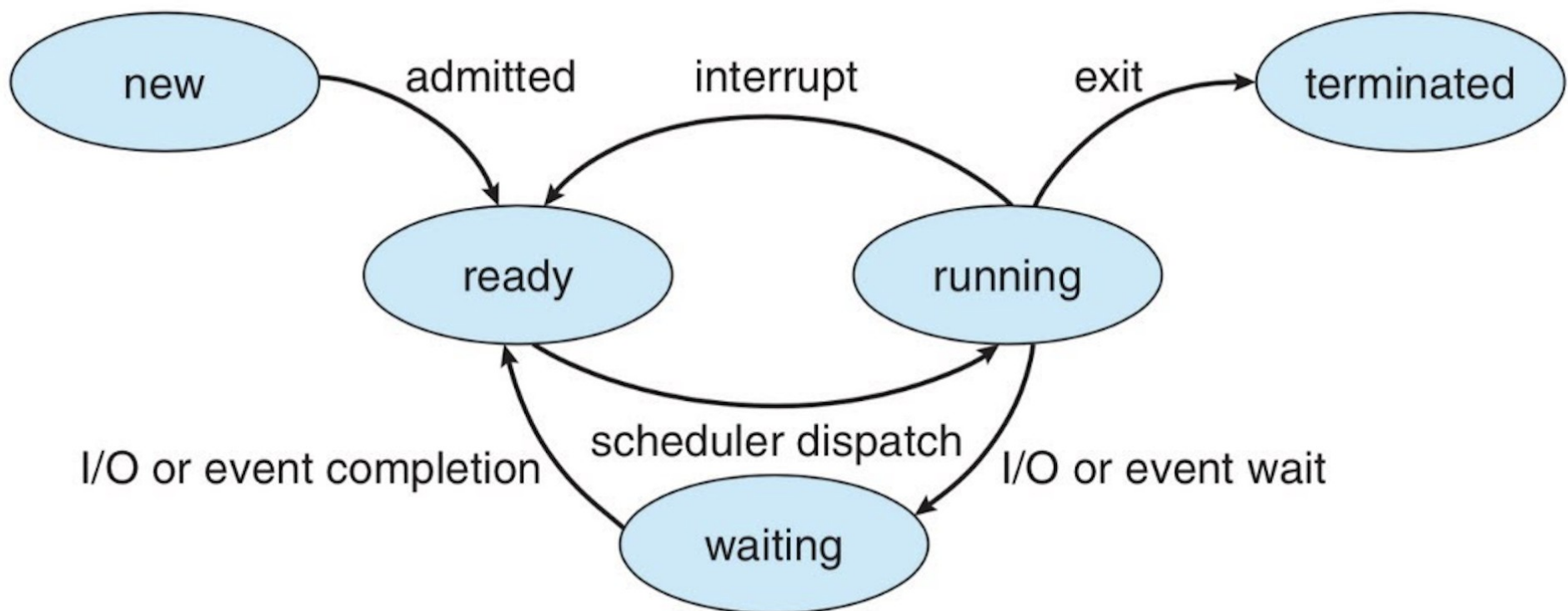
**Any Questions?**

# Purpose

- What's the purpose of assignment #1?
  - I've been asked this quite a few times
- What are we even doing?
  - To create a simple job dispatcher
    - Another program (*i.e. job.c*) will create jobs, and your program will dispatch the jobs based on some criteria
- What is the point of this?
  - *Next slide*
- What is the merit behind this?
  - *Next slide*

# Processes (1)

- *Recall from 2GA3:* Operating systems have scheduling algorithms and dispatchers to give processes CPU time



# Processes (2)

- Essentially, what you are doing is creating a job dispatcher/scheduler
  - The purpose of the job dispatcher is to decide which process gets resources (*i.e. CPU time*)
- The jobs are created by another program (*i.e. jobs.c*), and your program will decide which job gets priority (*i.e. CPU time*)
  - Signals are used to “start” jobs and “remove jobs from processing”
    - Note: Your program doesn’t literally do anything mentioned above. Rather it simulates it (*i.e. Pretend*)

**Now, Let's  
Start Coding**



# Coding (1)

- Before you start, read the description for Assignment #1
  - *Should I review it right now?*
- Anytime you see “**TODO**” and “**END TODO**”, you need to:
  - Read the description of what needs to be done
  - Implement/code it right after “**END TODO**”
  - *Walkthrough: Checking arguments in ``main()``*
- Recall that **assgn1.c** is the only file that needs to be modified
  - In other words, only **assgn1.c** has **TODOs**

# Coding (2)

- Note:
  - The “new” file I’ve provided does not give you any extra code
    - Rather, it explains in more detail what needs to be done
      - Everything has been outlined in a step-by-step fashion
    - Better correlation with the description on the course website
- Additional information on how to implement the required data structures are on the course website
  - Recall that you can use *some* code off the Internet, as long as you clearly state where you got it from
    - This will be useful for the Queue data structure

**THE**

**END**