# COSC 350 System Software Final Exam

05/07/2021

Name: _____

1. (20 pt.) Write syntax free bash scripts
   a) **task1a.sh**
      Write bash script which test each file names in current directory and display subdirectory names.

   b) **task1b.sh**
      Receive sequence of integers as command line arguments and calculate summation of integers and display the result
      Ex) for ./task2.sh 1 10 100 1000 10000
      Output: summation of arguments is 11111

2. (30 pt.) Write following error free compliable program.
   **task2a.c**
   - Create a message queue by using key value key = ftok(".", 'B')
   - Keep receiving two integer values from the command line until end of data (cnt-D)
   - You don't need validate data in task2a.c
   - Send each message to the message queue.

   **task2b.c**
   - Receive a message from the message queue which is created by task2a.c.
   - Check to make sure a message including two integer values.
   - Write error message if a message is not valid input.
   - Create a shared memory by using key value key = ftok(".", 'B')
   - Write two valid integers into the shared memory.
   - task2b.c will keep receiving data from the task2a until no more data

   **task2c.c**
   - Get two integer values from the shared memory which is created by **task2b**.
   - Calculate sum of two integers and display on screen.
   - **task2c** will keep running until no more new data in shared memory.

   When **task2a** is terminated, **task2b** and **task2c** must be terminated. Also, message queue and shared memory must be eliminated.

3. (15 pt.) Write a compilable C program (**task3.c**) in which a child process write a message "Hi, Mom" to a **file** named **foo**. The parent process reads the message from the file and prints it to standard output "My son said Hi Mom". Then send a message "what do you want" to the child through a **pipe**. Once the child receive a message from the parent, display on standard output "My Mom said what do you want". Assume that all system calls succeed (no need to error check). You must make sure a child process run first to write message in the file. **Use signal SIGUSR1** for synchronization between parent and child. **The file must be opened only one time before creating a child**. **Don't use vfork(), sleep(), alarm(), wait() or waitpid()** system calls.

4. (20 pt.) Write following error free compliable program. Assume that all system calls succeed (no need to error check for system calls).

   **task4a.c**
   - Create a semaphore with two members by using key value key = ftok(".", 'K')
   - Initiate both member of semaphore with value 1.

   **task4b.c**
   - Open existing semaphore created by the **task4a.c**
   - Create a child process. The parent and child process try run forever concurrently.
   - Both parent and child try to decrease both semaphore values one at a time.
   - Once both semaphore values are decreased (when both values become zero),
     - child process print "child is in critical section"
     - parent process print "parent is in critical section".
   - Then, parent and child try to increase both semaphore values one at a time.
   - Child count a number for each loop. Once count becomes 100, child send signal to parent which cause terminating the parent process.
   - After parent's termination, the child process removes the semaphore and terminate itself.

5. (15 pt.) Write a following syntax error free compliable C program (**task5.c**).
   - The parent creates a child process and try to run forever concurrently.
   - The child process just keep printing "child is running".
   - The parent process also keep printing "parent is running".
   - For each print, both parent and child need sleep one second.
   - After printing 10 times, parent send signal SIGUSR1 to child to terminate the child process.
   - When the child process gets a SIGUSR1 from the parent, the child process will print "I am terminated by my parents" and terminate itself.
   - Once a child is terminated, parent will print "My child is gone so I am now" then terminate itself. The parent must recognize the child's termination by signal.

How to submit.

- If a program is not compliable, let me know.
- Follow the file names
    - task1a.sh, task1b.sh
    - task2a.c task2b.c task2c.c
    - task3.c
    - task4a.c task4b.c
    - task5.c
- Submit by email to cosc350@gmail.com