Name: _____

UID: _____

Final Exam - CS 6013

University of Utah / School of Computing Apr 18, 2022

This is a take-home exam. When answering each question, you are allowed to use class materials (lectures, slides, textbook, etc.) as well as static on-line materials (e.g.: papers, tutorials, documentation, etc.). As applicable, you may use your computer to implement and test solutions. You may **NOT** copy answers / code from any of these sources. You may **NOT** work with anyone (classmates or others), share or receive answers from anyone, or discuss the questions with anyone except the professor or a TA – and in this case only to clarify a question if it is not clear.

The answers to the questions must be written in English using proper grammar, punctuation, etc., and be typed into a Word (or equivalent) document and submitted electronically by the due date.

Make sure to include your name and U Id at the top of your document.

Your answers should be as concise as possible, while thoroughly covering the question. They will be graded primarily on your ability to demonstrate understanding of the issue, and secondarily on how they are written / organized / etc. Unless otherwise stated, each answer should be approximately one page in length.

The Final Exam grade will be apportioned as follows: 70% on your written answers, 30% on an oral examination (with the TA) in which you will spend 10 minutes explaining in detail one of your answers (chosen by the TA). Note, if during the oral exam it becomes evident that you do not understand the answer you have written, you will receive 0 points.

If you do not understand a question, please ask for clarification.

1) Write a recursive function in assembly language to calculate the Nth Fibonacci number. If your function were to be called in C, it would look like:

```
int f = fibonnaci( 7 ); // f would be assigned the value 8.

N-th Position:   1, 2, 3, 4, 5, 6, 7, 8…

Fibonacci value: 0, 1, 1, 2, 3, 5, 8, 13…
```

Each line of your assembly code should be thoroughly documented explaining why you wrote it, and what purpose it has.

Some things to keep in mind: Because your function will be calling other functions (at least itself) 1) you can't rely on values in registers being the same when those function calls return, and 2) you will need to handle adjusting the stack pointer correctly. 3) You may assume that `num` will always be >= 1 when called from the C program.

Note, in C, fibonnaci would be written as:

```
long fib( long num )
{
   if( num == 1  ) {
      return 0;
   }
   else if( num == 2 ) {
      return 1;
   }
   return fib( num - 1 ) + fib( num - 2 );
}
```

2) Discuss the differences between threads and processes. Discuss process (thread) scheduling. Include a discussion on system calls (what they are, why they are used, and how they affect scheduling), including how they affect OS mode. Discuss what context switching is and how it relates to these things.

3) Explain the concept **spatial** and **temporal locality**. Why is it important that a developer understand these concepts? In your answer, make sure to address (in detail) the process a computer goes through to load data from main memory into the CPU. How is this effected by running multiple threads that process the same data (variable)?

4) Describe the difference between physical and virtual memory. How and why does this relate to paging and virtual addresses (and what parts of the CPU are involved in this process)? Discuss the implications of 32-bit vs 64-bit addresses. How is a "typical" virtual address space laid out? Why is this "idealized" picture not completely accurate in the age of threaded / 64-bit systems?

5) Discuss threading and different ways it can be used to speed up a program. Discuss critical sections – what they are, what problems they can cause, and ways to solve these problems. What is a mutex, condition variable, and barrier and why are they used? Write C++11 code that implements a barrier using mutexes and/or condition variables.