FI Concession Assessment
Due Date: 25 August 2015. No Late Submissions.

Instructions:

1) This FI concessions assessment opportunity is a portfolio of evidence. You must record your progress diligently using the GIT source control system. Failure to do so will result in a loss of marks.

2) This opportunity is afforded to you, and no one else but you. To this end, you must complete the form on the following page sign it, and submit it along with your portfolio. **Having another person complete the assignments contained as part of this document will result in disciplinary action, and potentially suspension or expulsion from the university.**

3) Read the questions very carefully and make sure you answer them fully.

4) This 7 pages document must be returned to us.

SM

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsmile: +27 429 12 429 4150
www.unisa.ac.za

The University of South Africa
College of Science Engineering and Technology
SCHOOL OF COMPUTING
STATEMENT OF AUTHENTICITY

Student Name: Sean Patrick McCann

PLAGIARISM is the presentation by a student of an assignment or piece of work which has in fact been copied in whole or in part from another student's work, or from any other source (eg published books or periodicals or material from Internet sites), without due acknowledgement in the text.

DECLARATION:

- I understand that the purpose of this portfolio is to assess **my** knowledge of the subject matter of the module COS3721, and as such I must present evidence to this fact.
- I therefore declare that all the work submitted as part of this portfolio is **my** own, and does not involve any form of plagiarism. I declare that I will not submit any work herein which was not conceived of, and **written by myself**.
- I acknowledge that I may make use of the COS3721 study material as presented in this module.

Student signature: Sean McCann

Date: 23 / 08 / 2015

SM

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsmile: +27 429 12 429 4150
www.unisa.ac.za

Table of Contents

SM

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsmile: +27 429 12 429 4150
www.unisa.ac.za

PROJECT DESCRIPTION

QUESTION I - Process Concept (SGG, Exo 3.7, P. 149)

Using the program in Figure 3.33, identify the values of pid at line A, B, C, and D. (Assume that the actual pids of the parent and child are 2600 and 2603, respectively.)

```
#include <sys/types.h>
#include <studio.h>
#include <unistd.h>

int main ( )
{
Pid_t pid, pid1;

        /* fork a child process */
        pid = fork ( );

        If (pid < 0 ) { /* error occurred */
                fprint (stderr, "Fork failed");
                return 1;
        }
        Else if (pid = = 0 ) { /* child process */
                pid1 = getpid ( );
                printf ("child: pid = %d", pid); /* A */
                printf ("child: pid1 = %d", pid1); /* B */
        }
        Else { /* parent process */
                pid1 = getpid( );
                printf("parent : pid = %d",  pid); /* C */
                printf("parent:  pide1 = %d, pid1) ; /* D */
        wait(Null) ;
        }

        return 0 ;
}
```

Figure 3.33 what are the pid values?

SM

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsmile: +27 429 12 429 4150
www.unisa.ac.za

QUESTION II - Multithreaded Programming (SGG, project 2, PP. 196-197)

Write a multithreaded sorting program that works as follows: A list of integers is divided into two smaller lists of equal size. Two separate threads (which we will term sorting threads) sort each sublist using a sorting algorithm of your choice. The two sublists are then merged by a third thread – a merging thread – which merges the two sublists into a single sorted list.

Because global data are shared across all threads, perhaps the easiest way to set up the data is to create a global array. Each sorting thread will work on one half of this array. A second global array of the same size as the unsorted integer array will also be established. The merging thread will then merge the two sublists into this second array. Graphically, this program is structured according to Figure 4.20.

This programming project will require passing parameters to each of the sorting threads. In particular, it will be necessary to identify the starting index from which each thread is to begin sorting. Refer to the instructions in Project 1 for details on passing parameters to a thread.

The parent thread will output the sorted array once all sorting threads have exited.

QUESTION III - Synchronisation (SGG, Exo 6.13, PP. 295-296)

A Multithreaded web server wishes to keep track of the number of requests it services (known as hits). Consider the two following strategies to prevent a race condition on the variable hits. The first strategy is to use a basic mutex lock when updating hits:

First strategy	Second strategy
Int hits; *Mutex_lock hit_lock;* *Hit_lock.acquire();* *Hits++;* *Hit_lock.release();*	*Atomic_t hits;* *Atomic_inc (&hits);*

Explain which of these strategies is more efficient.

SM

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsmile: +27 429 12 429 4150
www.unisa.ac.za

QUESTION IV - Memory-Management Strategies (SGG, Programming problem, P. 387)

Assume that a system has a 32-bit virtual address with a 4-KB page size. Write a C program that is passed a virtual address (in decimal) on the command line and have it output the page number and offset for the given address. As an example, your program would run as follows:

./a.out 19986

Your program would output:

The address 19986 contains:
Page number = 4
Offset = 3602
Writing this program will require using the appropriate data type to store 32 bits. We encourage you to use unsigned data types as well.

PROJECT TASKS

Question I	Analyse the program, identify the pid in line A, B, C and D. The answers expected from you are therefore the values of pid.	Type / written
Question II	We expect you to write a running program in C, (C++, C# or Java) that implements three different threads to sort a list of integers.	Program (ThreadProg)
Question III	We expect you to analyse the two strategies and explain in plain English which one more efficient than the other	Type / written
Question IV	We expect you to write a C program that determines from an input virtual address, in decimal, the page number and the offset	Program (SynchroProg)

SM

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsmile: +27 429 12 429 4150
www.unisa.ac.za

PROJECT SUBMISSION

Your submission will include: This document, the answer book, and your program.

1) This document

Please print the document, put your initial at the bottom of each page, fill-in the space for the student number at the header, complete your name on page 2 and sign at the space provided on the same page. The completed document must be scanned and sent back to us together with the answer book and the program.

2) Answer book

Please answer all the written questions on the same document. Include your initial at the bottom of each page and your student number in the space provided.

3) Programming

- Use the same interface for both programs (ThreadProg and SynchroProg). The user must be able to lunch the execution of any of the two programs from the common interface
- Compile the program, correct all the errors and make sure it runs properly
- Create a simple installer for your program
- Zip (e.g. WinRar) all the source code files, the tested executable file, as well as the installer and send to us.

MARKING

Rubrics			**Question Marks**	**Student mark**
Typed/ written	Question I		16 marks	
	Question III		14 marks	
Programming	Common Interface		10 marks	
	Definitions (variables)		10 marks	
	ThreadProg	sorting threads	20 marks	
		merging thread	10 marks	
	SynchroProg		20 marks	
Total			**100 mark**	

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsmile: +27 429 12 429 4150
www.unisa.ac.za