**CSE 430 Project 2 Part 2 Hints**

For this part of the project, the hints given in the skeleton code are not complete. You are provided with code that must be completed and must be placed in the appropriate module as required by your assignment. You must also write your own code to call and run the functions provided to you.

There are also some changes that are needed to be made to sched.h and fork.c system files and they have been provided for you at the end of this document.

This part of the project must be implemented as a module and run likewise. Do not implement it as a system call.

The functions needed to complete this part of the project, are listed in the supporting slides that were uploaded. You need to go through those functions and understand how they work and use them in the appropriate manner in the skeleton code provided.

Test cases have already been provided for you and you must use them to test your code prior to submission and demo.

Add the following lines to the end of the task\_struct located in include/linux/sched.h

**int fork\_count;**

**int descendant\_count;**

Add the following variables after the task\_struct has ended -

**extern volatile unsigned long fdb\_flag;**

**extern volatile unsigned int fdb\_fork\_threshold;**

**extern volatile unsigned int fdb\_descendant\_threshold;**

**extern volatile unsigned int fdb\_pid\_threshold;**

**extern struct task\_struct \*fdb\_suspect;**

**extern struct task\_struct \*hello\_kthread;**

**extern struct semaphore \*sem\_lock;**

Add the following to the kernel/fork.c file

1. include linux/semaphore.h

2. Locate the function called long\_do\_fork() and add the following line at the beginning

**struct task\_struct \*parent = NULL;**

3. Add the following toward the end of the long\_do\_fork() function

**p->descendant\_count = 0;**

**parent = current;**

**while(parent->pid > fdb\_pid\_threshold)**

**{**

**parent->descendant\_count++;**

**if(parent->descendant\_count > fdb\_descendant\_threshold**

**&& hello\_flag == 1)**

**{**

**fdb\_suspect = parent;**

**}**

**parent = parent->parent;**

**}**

**current->fork\_count++;**

**p->fork\_count = 0;**

**if (fdb\_suspect != NULL && hello\_flag == 1) {**

**up(sem\_lock);**

**}**