SOLUTION ASSIGNMENT 3 CMPT300 SUMMER 2016

You may choose to complete problem 1b) OR problem 4).

Problem 1a) will be graded.
Problems or parts of problems that add up to an additional 50 points will be graded.
You will not be told in advance which problems or parts of problems will be graded.

1. Consider Smaug's world as described in assignment 2.

Assume each of the following

- The first process in the simulation will spawn Smaug, and will spawn (over time) each of the cows, sheep, hunters and thieves.
- Sheep arrive (new sheep processes are created) at random times. The length of the interval between the arrivals of successive sheep is drawn from a uniform distribution. The length of interval n is the length of time between the arrival of sheep n and the arrival of sheep n+1.The distribution of possible interval lengths is uniform between 0 seconds and maximumSheepInterval seconds (0 probability of an arrival interval outside this range).
- Cows arrive (new cow processes are created) at random times. The length of the interval between the arrivals of successive cows is drawn from a uniform distribution. The distribution of possible interval lengths is uniform between 0 seconds and maximumCowInterval seconds (0 probability of an arrival interval outside this range).
- Similarly, treasure hunters arrive at random intervals drawn from a uniform distribution. The distribution of possible interval lengths is uniform between 0 seconds and maximumHunterInterval seconds (0 probability of an arrival interval outside this range).
- Similarly, thieves arrive at random intervals drawn from a uniform distribution. The distribution of possible interval lengths is uniform between 0 seconds and maximumThiefInterval or maximumHunterInterval seconds respectively (0 probability of an arrival interval outside this range).
- Smaug's treasure initially contains 400 jewels
- The probability that a particular treasure hunter or thief will not be defeated is winProb.
- Your program should request the values of maximumSheepInterval, maximumCowInterval, maximumHunterInterval, maximumThiefInterval and winProb from the user and use those values to initialize your simulation.

Your simulation should terminate when any one of the following conditions is met

Smaug has eaten 14 sheep and 14 cows
Smaug has defeated 12 treasure hunters or 15 thieves
Smaug has no treasure
Smaug has 800 jewels.

While your simulation is running, and after it terminates you must assure that no zombie (defunct) processes remain. At no time during execution should you have more than 5 zombies. You must also assure that all semaphores and mutexes allocated are released before your simulation terminates. To

check if there are semaphores or shared memory still allocated use the ipcs function in LINUX command line. Be sure to remove any leftover shared memory or mutex sets using ipcs (or the provided script) before running your simulation again. There are limited numbers of these resources available in the system and if none are available strange things begin to happen even in other applications. Restarting your host does not remove the leftover shared memory or semaphore sets, you must remove then yourself.

Your program should produce output (to the screen) each time something happens. That is when you run your program it should produce output that includes the following messages (in the order the actions these messages describe actually happen). Note comments in brackets are not part of the message to be printed.

Sheep with PID ….. is born
Cow with PID ….. is born
Treasure hunter with PID ….. has found the path
Thief with PID ….. has found the path
Treasure hunter with PID ….. has entered the valley
Thief with PID ….. has entered the valley
Sheep with PID ….. has been enchanted
Cow with PID ….. has been enchanted
…… sheep are now enchanted
….. cows are now enchanted
The last sheep (PID ….) in a snack wakes the dragon
The last cow (PID …..) in a snack wakes the dragon
Smaug is going to sleep
Smaug has been woken up
Smaug has discovered a snack
Smaug is eating a snack
Smaug is eating a cow
Smaug is eating a sheep

Sheep with PID …… has woken up to be eaten
Cow with PID …… has woken up to be eaten
Sheep with PID ….. dies
Sheep with PID ……. grazes for ……. ms
Cow with PID …… grazes for …… ms
Cow with PID …… dies
Cow with PID ….. has been eaten
Sheep with PID …. has been eaten
Treasure hunter with PID ….. has been defeated
Treasure hunter with PID ….. receives treasure
Smaug let a treasure hunter see his cave
Smaug is fighting a treasure hunter
Smaug is playing with a thief
Smaug's treasure includes …. jewels

Smaug defeats a thief
Smaug gives a thief jewels
Smaug takes a swim
Thief with PID ….. leaves
Treasure hunter with PID …. leaves
Smaug is playing with a thief
Smaug finished eating one sheep
Smaug finished eating one cow
Smaug has defeated a thief
Smaug has finished a snack (2 sheep processes and 2 cow process have been terminated)
Smaug has finished a battle (1 treasure hunter process/thread has been terminated)
Smaug has finished a game (1 thief process/thread has been terminated)
Smaug has added to his treasure he now has M jewels
Smaug has lost some treasure he now has M jewels
Smaug has no more treasure

You may add additional messages if you wish.

a) **[50 points]** Implement a C solution to this problem. You must use processes, shared variables, mutexes and semaphores. You may not use threads, or signals (other than kill when cleaning up at the end of the simulation, even kill should not be used during the simulation). You must use C libraries sem.h for semaphores and shm.h for shared memory. You must make a shared variable to hold a copy of each semaphore counter. Use that shared variable when you check semaphore values. Do not use semcntl() to directly check the semaphore values. You must protect each shared variable using a mutex.
   You may wish to start by implementing only Smaug and the sheep and their interactions. If you implement only these parts of Smaug's world you can earn up to 20 points. Next add the cow, and the interactions of the cow with the sheep and with Smaug for up to another 10 points. Add the thief and all interactions of the thief with Smaug, cows, sheep and Smaug's treasure for another 10 points. Finally add the hunter and the interactions of the hunter with Smaug, thieves, cows, sheep and Smaug's treasure for the final 10 points.
b) **[20 points]** Implement a C solution for the problem using threads and semaphores. Use only pthreads library and C library semaphore.h for threads and semaphores. You may not use multiple processes, or signals. For the bonus problem consider only Smaug and the cows and the sheep.

2. Consider a series of processes that are sharing a variety of resources. To avoid or recover from deadlocks it is necessary to be able to determine if a system is likely to deadlock. This decision may be made based on the 'state' of the system and the Banker's algorithm.
   a) **[10 points]** Consider the system with the state given below. In point form, explain what the terms in these equations represent in terms of the system resources and the processes presently running in the system.
   b) **[20 points]** Use the banker's algorithm to determine if the state given in C is a safe state for this system. For this state of the system can you state that any of the processes will deadlock. If you can state that processes will deadlock which process will deadlock?

$$R = \begin{bmatrix} 5 & 4 & 1 & 4 \\ 6 & 3 & 3 & 1 \\ 2 & 3 & 0 & 2 \\ 1 & 4 & 1 & 3 \\ 3 & 0 & 2 & 4 \\ 2 & 7 & 1 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 1 & 0 & 0 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 2 & 1 \end{bmatrix}$$

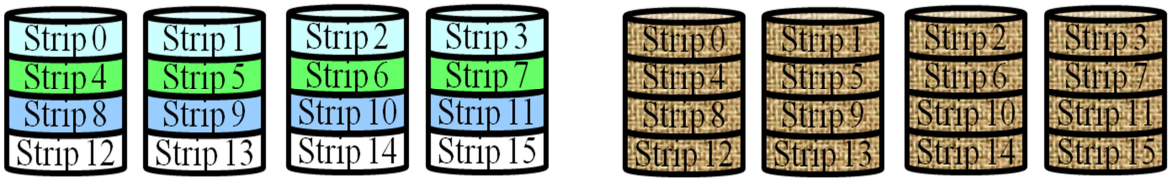
$$E = \begin{bmatrix} 6 & 8 & 4 & 6 \end{bmatrix}$$

3. Consider a hard disk. The disk has N=128 tracks and 24 sectors per track. The seek time per track is 1 ms. The rotation speed of the disk is 7200rpm.
   a) **[10 points]** Assume that once the head is in position above the appropriate track you need to wait an average of half a rotation of the disk before the head can read your data. What are the maximum and minimum rotational latency of this disk? What is the average rotational latency of this disk?
   b) The following sequence of disk access requests have been received by the computer systems. The numbers are the track numbers. Numbering starts at 0 (the centremost track on the disk).
      24, 68, 3, 17, 57, 49, 91, 121
      Determine the actual average seek length for this particular series of accesses based on each of the following algorithms. Assume the read head starts at track 48
      i. **[10 points]** Shortest seek first (move the fewest number of tracks to the next entry)
      ii. **[10 points]** Circular Scan (towards the outside of the disk first, direction of increasing track number)
      iii. **[10 points]** LOOK algorithm (towards the outside of the disk first, direction of increasing track number

c) **[20 points]** Assume that we have 8 of these disks, and we are using RAID 1 to combine these 8 disks. For this particular RAID system 1strip holds 1 track of data. The data for a particular file is loaded into tracks in the following order. Track 0 disk 0, Track 0 disk 1, Track 0 disk 2, Track 0 disk 3, Track 1 disk 0, Track 1 disk 1, … , Track 1 disk 3, …, Track 127 disk 0, …, Track 127 disk 3 and the same data is simultaneously loaded onto Track 0 disk 4, Track 0 disk 5, Track 0 disk 6, Track 0 disk 7, Track 1 disk 4, Track 1 disk 5, … , Track 1 disk 7, …, Track 127 disk 4, …, and Track 127 disk 7. (This is illustrated in the diagram below). The second 4 four disks are used to create a second copy of the data. Assume that the raid controller is capable of simultaneously accessing all disks. The raid controller contains eight buffers. Each buffer holds one track of data. Each of these buffers is reserved for the use of one of the eight disks. Assume that

i. Transferring one track of data from the disk, to one of the RAID controller's internal buffers takes 0.4 seconds
ii. A track of data must be completely loaded into the buffer of the RAID controller before it can be transferred to memory using the DMA
iii. All RAID controller buffers must be emptied (transferred to memory using the DMA) before any RAID controller buffer can be refilled.
iv. The DMA transfer rate (RAID controller buffer to memory is 4 GBytes per second (1 GByte is 2^{30} bytes)
v. The DMA setup time is negligible.
vi. One DMA transfer is used to transfer the contents of one buffer to memory
vii. The data being read is contiguous beginning at the start of Track 0 on platter 0 and filling a total of 14 tracks.
viii. Each track holds 60 Mbytes of data.

How long does it take to transfer the file from the disk to memory? How long would it take to transfer the file if the data were stored on track 0 to track 14 on disk 0 and disk 4? How would your answer differ if you were reading data from memory and writing it to the RAID 1 disks? When you answer each of these questions show step by step how you determined the length of time each transfer would take. Indicate in words what each quantity included in your calculated times represents.



4. **[20 points]** Consider a disk with N tracks numbered from 0 to N-1. Assume that requested sectors are distributed randomly and evenly over the disk.
   a) Calculate the probability of a seek of length j when the head is currently positions over track t. HINT: determine the total number of combinations, recognizing that all track positions for the destination of the seek are equally likely.
   b) Calculate the probability of a seek of length K
   c) Calculate the average number of tracks traversed by a seek HINT: Use the formula for expected value

$$E(x) = \sum_{i=0}^{N-1} i \sum \Pr[x = i]$$