John West
Cs314-002

10/9/16
HW3

1. (40 points) Implement semaphores using the pthread library. Consult the thread implemen- tation in DLXOS for inspiration (semaphores are implemented in synch.h and synch.c). Use pthread_mutex_lock and pthread_mutex_unlock (see man pthread mutex lock) to make the semaphore wait and post operations atomic. Use pthread_cond_wait (see man pthread cond timedwait, but don't use pthread cond timedwait) and pthread_cond_signal (see man pthread cond signal). Test your solution by implementing a simple rendezvous, e.g.:

```
THREAD A:
printf(``a1\n'');
sem_signal(aArrived);
sem_wait(bArrived);
printf(``a2\n'');

THREAD B:
printf(``b1\n'');
sem_signal(bArrived);
sem_wait(aArrived);
printf(``b2\n'');
```

See code in Zip file

2. (10 pts) Five jobs (A, B, C, D, and E), arrive in alphabetical order, and at almost the same time. They have estimated running times of 6, 4, 2, 6, and 8 minutes. Their priorities are 1, 2, 3, 4, and 5, respectively, with 5 being the highest priority. The processes arrive in the following quanta: 1, 1, 2, 4, 6, respectively. For each of the following scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead and assume a quantum of 1 minute. Show your work for partial credit.
    - Round Robin
    - Priority
    - FCFS
    - Shortest job

Round Robin:

Process	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A																										
B																										
C																										
D																										
E																										

Total time: 26 minutes
Average Time: 26/5=5.2 minutes

John West
Cs314-002

10/9/16
HW3

Priority Scheduling:

Process	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A																										
B																										
C																										
D																										
E																										

Total time: 26 minutes
Average Time: 26/5=5.2 minutes

FCFS:

Process	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A																										
B																										
C																										
D																										
E																										

Total time: 26 minutes
Average Time: 26/5=5.2 minutes

Shortest Job:

Process	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A																										
B																										
C																										
D																										
E																										

Total time: 26 minutes
Average Time: 26/5=5.2 minutes

3. (10 pts) The aging algorithm with a = ½ is being used to predict run times. The previous four runs, from oldest to most recent, are 10, 12, 20, and 4 msec. What is the sequence of predicted runtimes starting with the first of the observed runs (20), ending with the predicted runtime after the last observed run (12) .

   Observe 10, next predict: 10
   Observe 12, next predict: (20+12)/2 = 16
   Observe 20, next predict: (16+20)/2 =18
   Observe 12, next predict: (18+4)/2 =11

4. (10 pts) A real-time system needs to handle one voice call that runs every 10 msec and consumes 2 msec of CPU time per burst, plus a video at 40 frames/sec, with each frame requiring 15 msec of CPU time. Is this system schedulable? Explain.

   (1000/10) *2 + (40 *15) = 200 + 600 = 800. Yes, because it's in range of the scheduler.

5. (10 pts) Consider a swapping system in which memory consists of the following hole sizes in memory order: 8KB, 2KB, 20KB, 8KB, 6KB, 32KB, 24KB, and 8KB. Which hole is taken for successive segment requests of 4KB, 8KB, and 16KB when each of the following algorithms is used

   - First fit
   - Best fit
   - Next Fit
   - Worst Fit

   First Fit: 8, 20, 32
   Best Fit: 32
   Next Fit: 6,8,24
   Worst Fit: 20, 24, 32

6. (10 pts) For each of the following decimal virtual addresses, compute the virtual page number and offset for a 4KB page and for an 8KB page: 4097, 8193, 55555, 999999.

   4KB = 1024 bytes * 4=4096 bytes
   For 4097/4096 = 1.0002 = 1
   Offset: 4097-(1*4096) = 1

   8KB = 1024 bytes * 8=8192 bytes
   For 4097/8192 = 0.5=0
   Offset: 4097-(0*8192) = 4097

address	Page(4KB)	Offset(4KB)	Page(8KB)	Offset(8KB)
4097	1	1	0	4097
8193	2	1	1	1
55555	13	2307	6	6403
999999	244	575	122	575

7. (10 pts) A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are shown below (with times in clock ticks):

Page	Loaded	Last Ref.	R	M
0	100	220	1	1
1	90	210	0	1
2	140	200	1	0
3	110	190	0	1

NRU: Page 1 because R=0 & M=1
FIFO: page 1 because loaded at 90
LRU: page 3 because the last reference was 190
second chance replace: page 3 because it’s loaded at 110 and the reference bit is 0