Noor Thabit
CSCE-410
HW1

1. Problem 1
   A. child: pid = 0. Because the section is executed if (pid == 0).
   B. child: pid1 = 2603. Because pid1 = getpid() is executed within the child process block and the actual pid of the child is given, 2603.
   C. parent: pid = 2603
   D. parents: pid = 2600 Because pid1 = getpid() is executed within the parent process block.

2. Problem 2

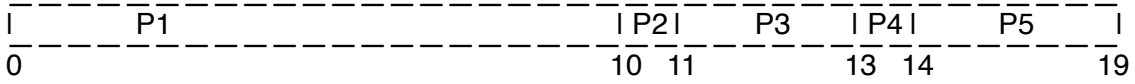
   In most of the cases is not better. The kernel is not aware of the user-level threads, which means the kernel can't assign process to different cores.

3. Problem 3

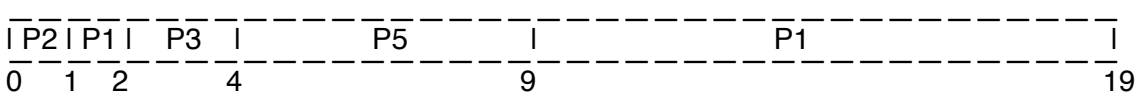
   CHILD: value = 5. Because *pthread_create(&tid,&attr,runner,NULL)* calls *runner()* in its arguments, which set *value = 5;*
   PARENT: value = 0. Because *runner(),* is not being called within the parents process section.

4. Problem 4

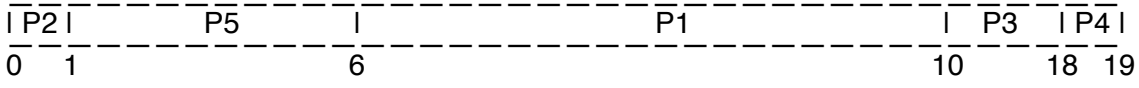
   A. **FCFS**



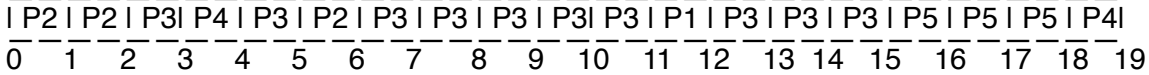
Preemptive SJF



Nonpreemptive priority



RR with quantum = 1.



B.

Process	FCFS	SJF	Nonp. priority	RR
P1	0	9	9	6
P2	10	0	0	0
P3	11	2	2	16
P4	13	1	1	18
P5	14	4	4	1

C.

Algorithm	Average
FCFS	9.6
SJF	3.2
Nonp. priority	8.2
RR	5.4

Problem 5:

A. Spinlocks are not appropriate for single-processor systems because in order for a process to break out of the spinlock, a different process has to be executing. Otherwise, the process will not leave off the processor voluntarily, so other processes do not get the chance to set the program condition required for the first process to make progress. On the other hand, in a multiprocessor system, other processes can be executing on other processors, so it get the chance to modify the program state in order to release the first process from the spinlock.
B. If a user-level program can disable interrupts, then it can disable the timer interrupt and prevent context switching from taking place. Consequently, it will use the processor without giving the chance for the other processes to execute.

Problem 6:

If two operations are trying to decrement a semaphore with a value of 1 non-atomically, then it is possible that both operations might proceed to decrement the semaphore value at the same time, which violates mutual exclusion.

Problem 7:

A.

1. **Mutual exclusion:** only one car can occupy each intersection at a time
2. H**old and wait:** cars can hold an intersection while waiting in a line for access to the next intersection
3. **No preemption:** an intersection that is occupied by a vehicle cannot be taken away from the vehicle unless the car is able to move.
4. **Circular wait:** the set of cars in the deadlock situation includes the cars in the middle of the intersection.

B.

Install traffic lights that only allow flow in one direction or the other at a time.
Allow a vehicle to cross an intersection only if it is assured that the vehicle will not need to wait whilst in the intersection (i.e. block the intersection). Not surprisingly, blocking an intersection is illegal even if you have the green light!

The traffic light should allow only one direction flow at anytime.
A vehicle is allowed to cross an intersection only if the vehicle will not need to wait whilst in the intersection.

Problem 8:

A.

	A B C D
P0	0 0 0 0
P1	0 7 5 0
P2	1 0 0 2
P3	0 0 2 0
P4	0 6 4 2

B. Yes