Operating Systems, 4th year Practical (15 Points) 1st Jan. 2015, 45 min.	Suez Canal University Faculty of Computers and Informatics Dr. Ahmed Sallam
Section: ----------	الاسم (باللغة العربية): ----------

This Exam in 2 pages, make sure to answer all the questions

1. (5 Points**) True (T) or False (F), and state Why?:**

a. When a process creates a new process using the fork() operation, they share the Stack segment. [x]

--New process with fork share nothing with its parent unless there is a defined shared memory ----------

b. In the Microkernel OS structure, everything below the system-call interface and above the physical hardware is the kernel. [x]

--In the monolithic OS structure----------

c. Switching the CPU to another process requires performing a state save of the current process and a state restore of a different process. This task is known as swapping. [x]

--Context switching ----------

d. Unlike spinlocks, semaphores can completely eliminate the busy waiting . [x]

---Semaphores only reduce the busy waiting as it still depending on a hardware solution----------

e. FIFO algorithm is an optimal scheduling algorithm with respect to the average waiting time. [x]

---SJF----------

2. (5 Points**) The following algorithm introduce a solution to the producer-consumer problem. However, this solution allowed at most "BUFFER_SIZE-1" items in the buffer at the same time.**

```
// Producer
while (true){
      while ((in+1)%BUFFER_SIZE== out);
      buffer[in] = next_produced;
      in = (in + 1) % BUFFER_SIZE;
}



// Consumer
while (true) {
      while (in==out);
      next_consumed = buffer[out];
      out = (out + 1) % BUFFER_SIZE;
}
```

Answer

```
counter=0;  //global variable
// Producer
while (true){
      while (counter==BUFFER_SIZE);
      buffer[counter] = next_produced;
      acquire();
      counter++;
      release();
}


// Consumer
while (true) {
      while (counter==0);
      next_consumed = buffer[counter-1];
      acquire();
      counter--;
      release();
}
```

a. Modify this algorithm to utilize the complete BUFFER_SIZE. (*note: use a pencil in case you need to re-modify your answer later*)

b. Do you think your modified algorithm will function correctly when the producer and consumer executed concurrently? Why?

---No, counter++ and counter-- represent race conditions and will not function properly if both were executed at the same time. ----------

---

c. If you answer (b) with "No", suggest another modification to the algorithm to function correctly when executed concurrently.

3. (5 Points **) Read the following program then answer the questions.**

```
1   #include <pthread.h>
2   #include <stdio.h>
3   #include <sys/types.h>
4   int x = 4;
5
6   void *runner(void *param) {
7     x = 12;
8     pthread_exit(0);
9   }
10
11  int main(int argc, char *argv[])
12  {
13    int y=8;
14    pid_t pid;
15    pthread_t tid;
16    pthread_attr_t attr;
17    pid = fork();
18
19    if (pid == 0) {
20        y=5;
21        pthread_attr_init(&attr);
22
23        pthread_create(&tid,&attr,runner,NULL);
24        pthread_join(tid,NULL);
25        printf("x= %d, y=%d\n", x, y);
26    }
27
28    else if (pid > 0) {
29        wait(NULL);
30        printf("x= %d, y=%d\n", x, y);
31    }
32
33  }//end main
34
```

a. Which API should be used to run this program?

----POSIX----------

b. How many threads would be created after executing this program?

----One----------

c. What would be the output from line 25?

-----x= 12, y=5----------

d. What would be the output from line 30?

-----x= 4, y=8----------

e. If we removed line 24, what would be the output from line 25?

-----x= 4, y=5------- I'll also consider x= 12, y=5-.However, it needs some tweaks to occur----------