Midterm Preparation

1. The most common secondary storage device is ____.
   A. random access memory
   B. solid-state disks
   C. tape drives
   D. magnetic disk

2. A ____ can be used to prevent a user program from never returning control to the operating system.
   A. portal
   B. program counter
   C. firewall
   D. timer

3. A(n) _____ is the unit of work in a system.
   A. process
   B. operating system
   C. timer
   D. mode bit

4. When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?
   A. The child process runs concurrently with the parent.
   B. The child process has a new program loaded into it.
   C. The child is a duplicate of the parent.
   D. All of the above

5. What is the correct order of operations for protecting a critical section using mutex locks?
   A. release() followed by acquire()
   B. acquire() followed by release()
   C. wait() followed by signal()
   D. signal() followed by wait()

6. What is the correct order of operations for protecting a critical section using a binary semaphore?
   A. release() followed by acquire()
   B. acquire() followed by release()
   C. wait() followed by signal()
   D. signal() followed by wait()

7. A(n) _____ refers to where a process is accessing/updating shared data.
   A. critical section
   B. entry section
   C. mutex
   D. test-and-set

8. How many philosophers may eat simultaneously in the Dining Philosophers problem with 5 philosophers?
   A. 1
   B. 2
   C. 3
   D. 5

9. _____ is/are not a technique for managing critical sections in operating systems.
   A. Peterson's solution
   B. Preemptive kernel
   C. Nonpreemptive kernel
   D. Semaphores

10. Multiprogramming of computer system increases
   A. Memory
   B. Storage
   C. CPU utilization
   D. Cost

11. What is a bootstrap program, and where is it stored?

12. Explain the concept of a context switch.

13. Distinguish between parallelism and concurrency.

14. What three conditions must be satisfied in order to solve the critical section problem?

15. What does this system call?

link(name1, name2)

16. Write a shell bash script to find and sort all txt files in the current directory.

17. Write the lock and unlock in the Consumer/producer with condition variables?

pthread_mutex_lock(& mut);

pthread_mutex_unlock(& mut);

```
pthread_mutex_t mut;
pthread_cond_t full, empty;
```

Producer thread:	Consumer thread:

while(1) {
item = produceItem();
while((in+1) % BUFF_SIZE == out){
pthread_cond_wait(& full,& mut);
}
buffer[in] = item;
in = (in + 1) % BUFF_SIZE;
count ++;
pthread_mutex_signal(& empty);
}
	while(1) {
	while(in == out){
	pthread_cond_wait(& empty,& mut);
	}
	item = buffer[out];
	out = (out+1) % BUFF_SIZE;
	count --;
	pthread_mutex_signal(& full);
	consumeItem(item);
	}

18. Assume a CPU instruction cycle with three stages: fetch, decode and execute. For every instruction, the fetch stage takes 2ns, the decode stage takes 2ns, and the execute stage takes 2ns.

A. How many instructions per second can this CPU execute on average if the stages are not parallelized?
B. How many instructions per second can this CPU execute on average if all stages are operating in parallel?

19. What is the output of the following code?

```
int p;
p = fork();
p = fork();
p = fork();
p = fork();

if (p==0)
  cout<<"HELLO"<<endl;
```

20. What is the output of the following code?

```
for(int i=1; i<=3; i++)
  fork();

cout<<"Hello";
```

1	D
2	D
3	A
4	D
5	B
6	C
7	A
8	B
9	A
10	C
11	A bootstrap program is the initial program that the computer runs when it is powered up or rebooted. It initializes all aspects of the system, from CPU registers to device controllers to memory contents. Typically, it is stored in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), known by the general term firmware, within the computer hardware.
12	Whenever the CPU starts executing a new process, the old process's state must be preserved. The context of a process is represented by its process control block. 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 a context switch. When a context switch occurs, the kernel saves the context of the old process in its PCB and loads the saves context of the new process scheduled to run.
13	A parallel system can perform more than one task simultaneously. A concurrent system supports more than one task by allowing multiple tasks to make progress.
14	In a solution to the critical section problem, no thread may be executing in its critical section if a thread is currently executing in its critical section. Furthermore, only those threads that are not executing in their critical sections can participate in the decision on which process will enter its critical section next. Finally, a bound must exist on the number of times that other threads are allowed to enter their critical state after a thread has made a request to enter its critical state.
15	create a file link name2 pointing to name1
16	script

17	**Producer thread:**
	
	while(1) {
	item = produceItem();

	pthread_mutex_lock(& mut);
	while((in+1) % BUFF_SIZE == out){
	pthread_cond_wait(& full,& mut);
	}
	buffer[in] = item;
	in = (in + 1) % BUFF_SIZE;
	count ++;
	pthread_mutex_signal(& empty);
	pthread_mutex_unlock(& mut);
	}
	Consumer thread:
	
	while(1) {
	pthread_mutex_lock(& mut);
	while(in == out){
	pthread_cond_wait(& empty,& mut);
	}
	item = buffer[out];
	out = (out+1) % BUFF_SIZE;
	count --;
	pthread_mutex_signal(& full);
	pthread_mutex_unlock(& mut);
	consumeItem(item);
	}

19	Hello * 8
20	Hello * 8