[CE3002] Operating System -- Homework 01

1. Choice of Questions (40%)

1. Which of the following would lead you to believe that a given system is an SMP-type system?

- [ ] A) Each processor is assigned a specific task.
- [ ] B) There is a boss–worker relationship between the processors.
- [x] C) Each processor performs all tasks within the operating system.
- [ ] D) None of the above

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
- [x] D) timer

3. Two important design issues for cache memory are ____.

- [ ] A) speed and volatility
- [x] B) size and replacement policy
- [ ] C) power consumption and reusability
- [ ] D) size and access privileges

4. What statement concerning privileged instructions is considered false?

- [x] A) They may cause harm to the system.
- [ ] B) They can only be executed in kernel mode.
- [ ] C) They cannot be attempted from user mode.
- [ ] D) They are used to manage interrupts.

5. The two separate modes of operating in a system are

- [ ] A) supervisor mode and system mode
- [ ] B) kernel mode and privileged mode
- [ ] C) physical mode and logical mode
- [x] D) user mode and kernel mode

6. If a program terminates abnormally, a dump of memory may be examined by a ____ to determine the cause of the problem.

- [ ] A) module
- [x] B) debugger
- [ ] C) shell
- [ ] D) ontrol card

7. Policy ____.

- [ ] A) determines how to do something
- [x] B) determines what will be done
- [ ] C) is not likely to change across places
- [ ] D) is not likely to change over time

8. A microkernel is a kernel ____.

```
- [ ] A) containing many components that are optimized to reduce resident memory size
- [ ] B) that is compressed before loading in order to reduce its resident memory size
- [ ] C) that is compiled to produce the smallest size possible when stored to disk
- [x] D) that is stripped of all nonessential components
```

9. To the SYSGEN program of an operating system, the least useful piece of information is _____.

```
- [ ] A) the CPU being used
- [ ] B) amount of memory available
- [x] C) what applications to install
- [ ] D) operating-system options such as buffer sizes or CPU scheduling algorithms
```

10. _____ provide(s) an interface to the services provided by an operating system.

```
- [ ] A) Shared memory
- [x] B) System calls
- [ ] C) Simulators
- [ ] D) ommunication
```

二、問答題 (60%)

11. What is the purpose of interrupts? (4%) What are the differences between a trap and an interrupt? (4%)

The purpose of interrupts in a computer system is to allow the processor to pause its current activities and immediately respond to an urgent event. Interrupts can be triggered by both hardware (like pressing a key on the keyboard) and software (like a running program encountering an error).

The difference between a trap and an interrupt lies in their source and purpose. An interrupt is typically caused by external events (like hardware devices), while a trap is caused by the execution of a specific instruction in the program. Traps are used to call operating system routines and to catch arithmetic errors in the program. In essence, an interrupt is asynchronous, triggered by external events, while a trap is synchronous, occurring because of the program's flow of control.

12. Describe three general methods for passing parameters to the operating system. (6%)

1. **Passing parameters in registers**: Some system calls pass parameters in registers. This method is efficient as it avoids memory access, but it is limited by the number of registers available.
2. **Passing parameters in a block of memory**: Some system calls pass parameters by storing them in a block of memory and passing the address of the block to the kernel. This method is flexible and can handle a large number of parameters, but it requires memory access.
3. **Passing parameters on the stack**: Some system calls push parameters onto the stack, which are then popped off by the operating system. This method is simple and flexible, but it may limit the number or length of parameters passed.

13. What are the two models of interprocess communication? (3%) What are the strengths and weaknesses of the two approaches? (3%)

1. The two models of interprocess communication are message-passing model and the shared-memory model.
2. Strengths:
    1. Message passing is useful for exchanging smaller amounts of data, because no conflicts need be avoided.
    2. Shared memory allows maximum speed and convenience of communication, since it can be done at memory transfer speeds when it takes place within a computer.
3. Weaknesses:
    1. Message passing can be slower than shared memory, as it requires copying data between processes.
    2. Shared memory can be more complex to implement and manage, as it requires synchronization mechanisms to avoid conflicts between processes.

14. Describe the differences among short-term, medium-term, and long-term scheduling. (5%)

- **Short-term scheduling**: Short-term scheduling, also known as CPU scheduling, involves selecting a process from the ready queue and allocating the CPU to that process. It determines which process will execute next and is typically based on criteria like priority, time quantum, or other scheduling algorithms.
- **Medium-term scheduling**: Medium-term scheduling involves swapping processes between main memory and disk storage. When a process is swapped out of memory, it is placed in a suspended state on disk, and when it is swapped back in, it is placed in the ready queue. This helps manage memory resources and can improve system performance.
- **Long-term scheduling**: Long-term scheduling involves selecting processes from the job pool and loading them into memory to create the ready queue. It determines which processes will be admitted to the system and can help balance system load and resource utilization.

15. Including the initial parent process, how many processes are created by the program shown in Figure 3.32? (6%)

```c
// Figure 3.32 How many processes are created?
#include <stdio.h>
#include <unistd.h>

int main() {
    int i;
    for (i = 0; i < 4; i++)
        fork();
    return 0;
}
```

- 16 processes are created by the program shown in Figure 3.32. The initial call of `fork()` will create the first out of 4 children of this parent, and each of the first children will have 3 remaining `fork()` calls. Which means that each will create 3 children, and the first child of each of those 3 will create 2, and so on.

16. Which of the following components of program state are shared across threads in a multithreaded process? (4%)

- [ ] A) Register values
- [x] B) Heap memory
- [ ] C) Global variables
- [ ] D) Stack memory

Heap memory and Global variables shared across in multi-thread process. In multi-thread process, each thread has its own stack and register values.

17. Consider the following code segment:

```c
pid_t pid;

pid = fork();
if (pid == 0) { /*child process*/
    fork();
    thread_create( ... );
}
fork();
```

- a. How many processes are created? (include main process) (4%)
  - 6 processes are created.
- b. How many threads are created? (4%)
  - 8 threads are created.

18. Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst Time	Priority

Process	Burst Time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5 all at time 0.

- a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1). (4%)
- b. What is the turnaround time of each process for each of the scheduling algorithms in part a? (4%)
- c. What is the waiting time of each process for each of these scheduling algorithms? (4%)
- d. Which of the algorithms results in the minimum average waiting time (over all processes)? (2%)

FCFS Scheduling Algorithm

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	2	2	2	0
P2	0	1	3	3	2
P3	0	8	11	11	3
P4	0	4	15	15	11
P5	0	5	20	20	15

SJF Scheduling Algorithm

Process	Arrival Time	Burst Time	Priority	Completion Time	Turnaround Time	Waiting Time
P1	0	1	1	1	1	0
P2	0	2	2	3	3	1
P3	0	4	2	7	7	3
P4	0	5	3	12	12	7
P5	0	8	4	20	20	12

Average Waiting Time

To calculate the average waiting time for each scheduling algorithm, we sum up the waiting times of all processes and divide by the number of processes.

For FCFS: (0 + 2 + 3 + 11 + 15) / 5 = 6.2 For SJF: (0 + 1 + 3 + 7 + 12) / 5 = 4.6 For nonpreemptive priority: (0 + 1 + 3 + 7 + 12) / 5 = 4.6

Therefore, both the SJF and nonpreemptive priority scheduling algorithms result in the minimum average waiting time of 4.6.

19. Which of the following scheduling algorithms could result in starvation? (3%)

- [ ] A) First-come, first-served
- [x] B) Shortest job first
- [ ] C) Round robin
- [ ] D) Priority

Shortest job first and priority-based scheduling algorithms could result in starvation.