

## Assignment 3

Course: Operating System

Semester: BSIT-V(A)

Deadline: 12 Nov, 2024

### Instructions:

The purpose of this assignment is to **build an understanding and practice** on course material. It is expected that students will **solve the assignment themselves** using attached lectures and books at Google classroom.

- Plagiarism will be straight marked as zero
- Copying straight from the internet will be considered as plagiarism.
- Submit assignment in pdf format.
- **No submission accepted after the due date and time.**

1. Keeping in mind the various definitions of operating system, consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answers.
2. How multi-programmed systems are used to get maximum utilization of CPU and I/O devices?
3. How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security)?
4. Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?
5. Distinguish between the client–server and peer-to-peer models of distributed systems.
6. What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?
7. How interrupts improve the efficiency of the processor and make use of a diagram to show how processor switches between different user programs and I/O programs during an interruption.
8. You have been providing technical services to Computer Science labs on a regular basis and you notice that their systems are very slow. Using your technical experience, you checked the printing queue and realized that they are 100 jobs waiting to be printed by a shared printer. You quickly checked the processor utilization metrics and noticed that it was idling and at the same time, a printer was busy printing papers. It came to your attention that processor was idling because it was waiting for the printer to finish printing before executing the next job. In your observation, what do you think is the problem? How would you improve processor utilization?
9. What exactly do you mean by Context in context switching?
10. What impact does context switching have on CPU performance, and how can scheduling algorithms minimize the overhead in a multi-tasking environment.
11. Why do we call a program passive entity and a process active entity?
12. Explain the purpose of system calls and how do system calls relate to the OS and to the concept of dual mode (user mode and kernel mode) operation.
13. Explain Windows and Unix system calls with examples.

14. Briefly explain the purpose of the command interpreter. Why is it usually separate from the kernel?
15. Why do some systems store the operating system in firmware, while others store it on disk?
16. Contrast and compare an application programming interface (API) and an application binary interface (ABI).
17. How are iOS and Android similar? How are they different?
18. Explain the use of fork () and pipe () system calls with the help of an example. Also, write the reasons of failure of these systems.
19. How parent and child share the address space (memory) after fork?
20. Mention any two differences between logical and physical memory.
21. What do you mean by the term “Dispatch Latency”?
22. (a) A database administrator is testing a connection pooling mechanism for a web application. The administrator creates a main process representing the database server and **initiates** several child processes to simulate multiple client connections. The database will allow access if at least two clients can connect successfully. The administrator wrote the following code to simulate the connection requests. Analyze how many times Connection Request 1 and Connection Request 2 will be logged as successfully established.

```
int main () {
    if (fork () && fork ()) {
        printf ("Connection Request 1 logged successfully!");
    }
    fork(); fork(); fork();
    printf("Connection Request 2 logged successfully!");
}
```

- (b) Differentiate between fork () and exec () system calls with the help of an example.
23. When a process creates a new process using the fork () operation, which of the following states is shared between the parent process and the child process?
  - a. Stack
  - b. Heap
  - c. Shared memory segments
24. Explain what the output will be at LINE A

```
int value = 5;
int main()
{
    pid_t pid;
    pid = fork();
    if (pid == 0) {                               /* child process */
        value += 15;
        return 0;
    } else if (pid > 0) {                          /* parent process */
        wait(NULL);
        printf("PARENT: value = %d",value);      /* LINE A */
        return 0;
    } }
}
```

25. Describe the actions taken by a kernel to context-switch between processes.
26. Explain the role of the init (or systemd) process on UNIX and Linux systems in regard to

process termination.

27. What will be the output of the given code?

```
int main()
{
    fork();
    fork() && fork() || fork();
    fork();

    printf("forked\n");
    return 0;
}
```

28. Give an example of a situation in which ordinary pipes are more suitable than named pipes and an example of a situation in which named pipes are more suitable than ordinary pipes.
29. Consider the RPC mechanism. Describe the undesirable consequences that could arise from not enforcing either the “at most once” or “exactly once” semantic. Describe possible uses for a mechanism that has neither of these guarantees.
30. Explain in detail what the PCB is and why it is crucial for process management. Additionally, describe the key pieces of information stored in a PCB.
31. Imagine you are a university student working on a group presentation. Each member is responsible for different sections (e.g., research, slides, speaking notes). As you manage your tasks, think of each member (including yourself) as a separate process in a computer system. Using this scenario, explain the states of a process (New, Ready, Running, Waiting, Terminated) in the process lifecycle. Include a diagram showing these states and transitions.
32. What resources are used when a thread is created? How do they differ from those used when a process is created?
33. What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?
34. Provide three programming examples in which multithreading provides better performance than a single-threaded solution.
35. What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?
36. Using Amdahl’s Law, calculate the speedup gain of an application that has a 60 percent parallel component for (a) two processing cores and (b) four processing cores.
37. You are part of a team developing an online food delivery platform. The platform handles multiple tasks, such as placing orders, updating delivery statuses, processing payments, and sending notifications to users. Sometimes, the system seems to switch between these tasks rapidly, but it does not always process them at the same time. For instance, when a customer places an order, the system updates the inventory first, then processes the payment, and finally sends a confirmation message. While these tasks happen quickly, they are not all processed at once.  
Based on the scenario, explain whether the platform is using concurrency, parallelism, or both. Justify your answer by describing how the tasks (e.g., order placement, payment processing) are being handled.
38. Imagine a shared bank account with a balance of \$100. Both Person A and Person B have access to this account and initiate withdrawals at almost the same time:  
Person A checks the account balance, sees \$100, and proceeds with their withdrawal of \$50.

Around the same time, Person B also checks the balance, sees \$100, and proceeds with their withdrawal of \$70.

- a) Based on the situation, what would be the final balance in the account if both withdrawals are processed without any synchronization mechanism (like locks)?
  - b) Explain why this situation creates a race condition and what could be done in an operating system to prevent this type of issue.
39. In the context of InterProcess Communication (IPC), under what circumstances would you choose to use message passing instead of shared memory, and vice versa? Discuss the implications of using each method in both distributed systems and on the same machine.
40. Which of the following scheduling algorithms could result in starvation?  
a. First-come, first-served      b. Shortest job first      c. Round robin      d. Priority
41. What does it mean to preempt a process?

\*\*\*\*\*

*Success is the result of preparation, hard work, and learning from failure.*