

CS-2006: Operating Systems

Saturday, 26th September, 2023

Course Instructors

Maryam Shahbaz, Muhammad Aadil Ur Rehman

Sections - A,B,C,D

Serial No:

Sessional Exam-I

Total Time: 1 Hour

Total Marks: 35

Student Name

Roll No.

Course Section

Student Signature

Signature of Invigilator

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space, write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have **Ten (10)** different printed pages including this title page. There are a total of **4** questions.
5. Calculator sharing is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

	Q-1	Q-2	Q-3	Q-4	Total
Marks Obtained					
Total Marks	15	7	2	11	35

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

Question 1 [15 Marks]

MCQs Answer Sheet

MCQs Number	Answer	MCQs Number	Answer
1	c	16	c
2	d	17	a
3	d	18	c
4	c	19	b
5	d	20	b
6	c	21	c
7	b	22	c
8	b	23	b
9	c	24	d
10	b	25	c
11	b	26	b
12	d	27	a
13	a	28	c
14	d,e	29	b,c
15	c	30	d

Mention the correct option label in the box provided in MCQs answer sheet on **page 2**.

Instructions:

- A. Overwritten answers will not be considered.
- B. The answers marked on MCQs Answer Sheet will be considered only.

An honest zero is better than a stolen one hundred

1. The CPU catches the interrupt and ____ it to the interrupt handler
 - A) raises
 - B) catches
 - C) dispatches
 - D) clears
2. In operating system, each process has its own ____
 - a) address space and global variables
 - b) open files
 - c) pending alarms, signals and signal handlers
 - d) all of the above
3. A process stack does not contain ____
 - a) Function parameters
 - b) Local variables
 - c) Return addresses
 - d) PID of child process
4. A Process Control Block(PCB) does not contain which of the following?
 - a) Code
 - b) Stack
 - c) Bootstrap program
 - d) Data
5. The state of a process is defined by ____
 - a) the final activity of the process
 - b) the activity just executed by the process
 - c) the activity to next be executed by the process
 - d) the current activity of the process
6. Which of the following does not interrupt a running process?
 - a) A device
 - b) Timer
 - c) Scheduler process
 - d) Power failure

7. Suppose that a process is in “Blocked” state waiting for some I/O service. When the service is completed, it goes to the _____.
 - a) Running state
 - b) Ready state
 - c) Suspended state
 - d) Terminated state

8. What is Interprocess communication?
 - a) allows processes to communicate and synchronize their actions when using the same address space
 - b) allows processes to communicate and synchronize their actions
 - c) allows the processes to only synchronize their actions without communication
 - d) None of the above

9. Which of the following component does not belong to PCB (Process Control Block)?
 - a) CPU registers
 - b) CPU scheduling information
 - c) Process queue information
 - d) Accounting information
 - e) None of the above

10. Which of the following statements is correct about virtual memory?
 - a) It is a combination of the logical-memory and physical-memory
 - b) It is a separation of user logical memory and physical memory
 - c) It is a virtual network memory
 - d) None of the above
 - e) All of the above

11. What is the use of directory structure in the operating system?
 - a) The directory structure is used to solve the problem of the network connection in OS.
 - b) It is used to store folders and files hierarchically.
 - c) It is used to store the program in file format.
 - d) All of the above
 - e) None of the above

12. The PCB is identified by _____.
 - a) Real-Number
 - b) Binary Number
 - c) Store block
 - d) Integer Process ID
 - e) All of the above

13. Which of the following methods is used to improve the main memory utilization?
 - a) Swapping
 - b) Operating system
 - c) Memory stack
 - d) None of these.

14. Which of the following is not typically a parameter passed to a file-related system call in UNIX-like operating systems?
 - a) File descriptor
 - b) File path
 - c) File access mode
 - d) Process ID
 - e) Pipe

15. What is the primary purpose of a system call in an operating system?
 - a) To execute user-level applications
 - b) To allow communication between different processes
 - c) To request services from the operating system kernel
 - d) To manage hardware resources
 - e) None of the above

16. System calls related to inter-process communication (IPC) are categorized under which type of system calls?
 - a) Process control calls
 - b) File system calls
 - c) Communication calls
 - d) Device control calls
 - e) None of the above

17. In the context of debugging, what does a breakpoint refer to?
 - a) A pause in the execution of a program at a specific line of code
 - b) A compilation error in the code
 - c) A memory leak in the program
 - d) A hardware failure in the computer
 - e) None of the above

18. Which IPC communication method is considered an example of direct communication?
 - a) Pipes
 - b) Message Queues
 - c) Shared Memory
 - d) Sockets
 - e) Signals

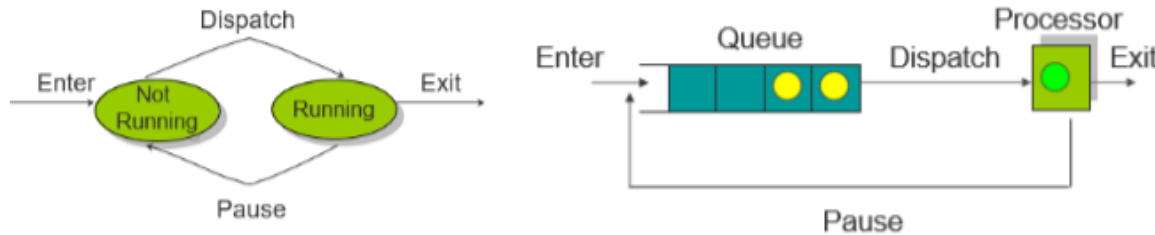
19. In Message Passing, what is the primary purpose of a mailbox or message queue?
 - a) To store messages for future retrieval by the sender.
 - b) To buffer messages temporarily during transmission.
 - c) To establish network connections between processes.
 - d) To execute non-blocking communication.
 - e) None of the above defines the primary purpose of mailbox or message

20. From a user point of view, which component of an operating system directly interacts with the user, allowing them to issue commands and run applications?
- a) Kernel
 - b) Shell
 - c) Scheduler
 - d) File System
 - e) None of the above
21. From a system perspective, which of the following tasks is typically not a responsibility of the operating system?
- a) Managing memory allocation and deallocation
 - b) Scheduling processes for execution
 - c) Providing a graphical user interface (GUI)
 - d) Handling file system operations
 - e) None of the above
22. From a user perspective, a computer system might appear to be idle while a system perspective reveals that background tasks are running. Which operating system component is responsible for managing these background tasks and ensuring that they don't disrupt user interactions?
- a) Kernel
 - b) Shell
 - c) Scheduler
 - d) File System
 - e) None of the above
23. Consider a single process running on a computer system that needs to perform an output operation, such as writing data to a file. Which of the following describes the typical interrupt timeline for this process?
- a) Process initiates the output operation, and an interrupt occurs immediately.
 - b) Process initiates the output operation, and the CPU continues executing other instructions until the operation is complete.
 - c) Process initiates the output operation, and the CPU is paused until the operation is complete.
 - d) Process initiates the output operation, and an interrupt occurs only after the operation is complete.
 - e) None of the above
24. When an “**exec**” system call is invoked during the process execution, it overlay's the current process, which of the following statements can be used to return it back to the originating process.
- a) Using wait system call
 - b) Using waitpid system call
 - c) Using fork() and wait() combination
 - d) It cannot be returned to the originating process.
 - e) It returns automatically to the code line where it was called
25. When a user application running in user mode requires a service from the operating system, which mechanism is typically used to transition from user mode to kernel mode?
- a) Traps
 - b) Interrupts
 - c) System calls

- d) Context switches
 - e) Signals
26. In a multi-mode operating system, what is the main advantage of having multiple modes of operation?
- a) Enhanced system performance
 - b) Improved system security
 - c) Greater user accessibility
 - d) Simplified system architecture
 - e) None of the above
27. During the boot process of a computer system, what is the first program that is loaded into memory and executed by the CPU?
- a) BIOS (Basic Input/Output System)
 - b) Operating System Kernel
 - c) Bootloader
 - d) Device Drivers
28. In a Unix-like operating system, a process uses the `fork()` system call to create a new child process. The child process inherits a copy of the parent's memory space, including data and code. Now, the parent process and the child process both exist. What is the primary difference between the two immediately after the `fork()` system call?
- a) The parent process executes the same code, while the child process executes a different code segment.
 - b) Both the parent and child processes execute the same code at the same memory addresses.
 - c) The child process executes a separate copy of the parent's code and data.
 - d) The parent process is suspended, and only the child process executes.
 - e) The parent process continues, whereas the child process is blocked until parent finishes its execution
29. Which of the following scheduler controls the degree of multiprogramming?
- a) Short-term Scheduler
 - b) Long-term Scheduler
 - c) Mid-term Scheduler
 - d) Kernel Scheduler
 - e) None of the above
30. In Unix-like systems, when a child process is created using `fork()`, which of the following statements is true about the relationship between the parent and child processes?
- a) They share the same Program Counter (PC).
 - b) They have the same PCBs.
 - c) They share the same heap memory.
 - d) They have different Process IDs (PIDs).
 - e) None of the above

Question 2[7 Marks]

According to Two State Process Model, at any given time a process is either Running or Not Running. The Processes that are Not Running at a particular time should be kept in a queue. Considering this context, answer the following questions:



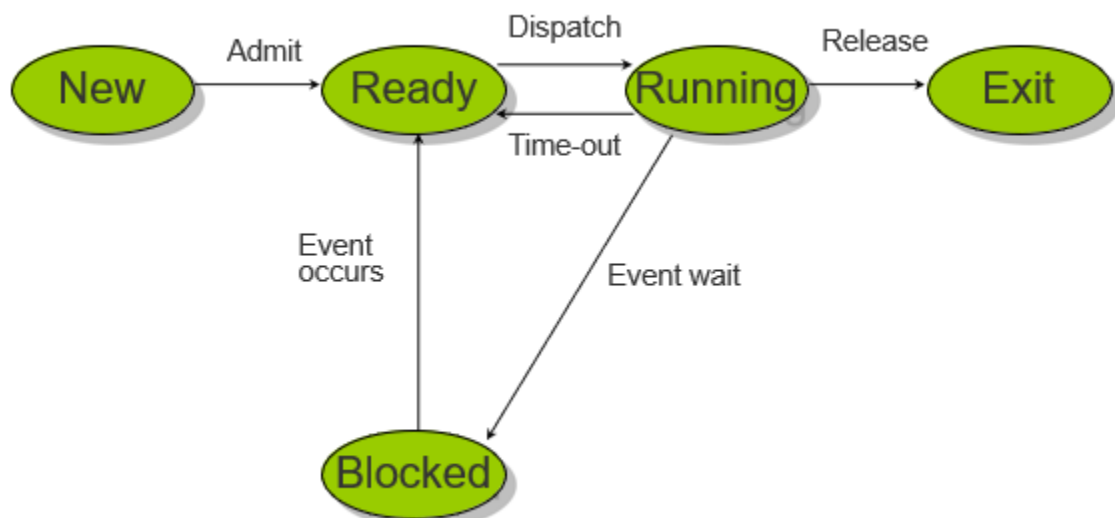
- A. Write the main issue in a two state process model. (at most 2 lines answer, otherwise, marks will be deducted) [1 Marks]

The main issues in a two-state process model is the presence of a single queue for both ready and blocked processes, which can lead to inefficient scheduling and resource utilization.

- B. What is the solution to the issue mentioned above in part A, that was discussed in the class? (at most 1 line answer, otherwise, marks will be deducted) [1 Marks]

The solution to the issue in a two-state process model is to implement a multi-level queue or multiple queues to differentiate between processes ready to execute and those waiting for I/O or other events, enabling more efficient scheduling and resource allocation.

- C. Draw a diagram representing the Five-State Process Model. Label each state and indicate the transitions between states that a process can go through during its lifecycle. [3 Marks]

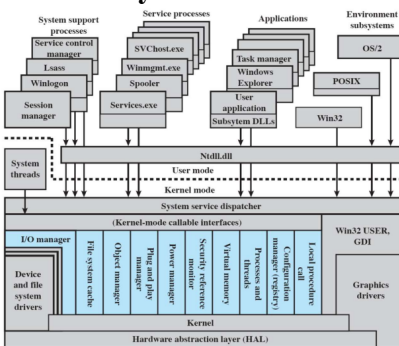


D. Name any 2 type of queues used in 5 state model , and give purpose of each queue. [2 Marks]

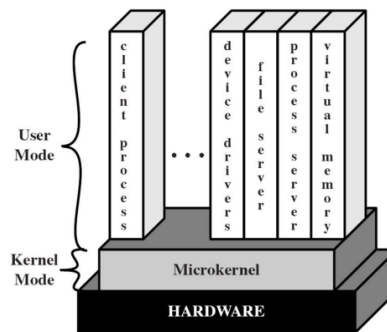
Queue Type	Purpose
Ready Queue	The purpose of the ready queue is to store processes that are ready to execute but waiting for CPU time, allowing for efficient process scheduling and execution.
I/O/Event Queue (or Device Queue):	The purpose of the I/O queue is to store processes that are waiting for I/O operations to complete, ensuring orderly access to I/O devices and preventing unnecessary CPU waiting.

Question 3[2 Marks]

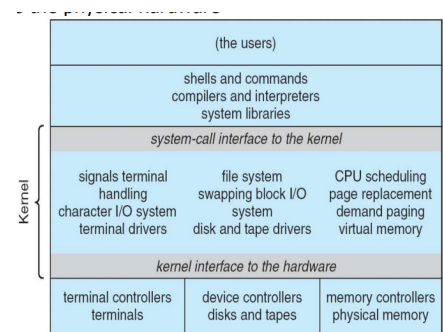
The diagrams of different operating system structures are provided below, label the diagrams, also identify which one of these is Linux/Unix structure.



**Modified Microkernel/
Windows 2000**



Microkernel



**Hybrid(layered/Monolithic)
Unix Architecture**

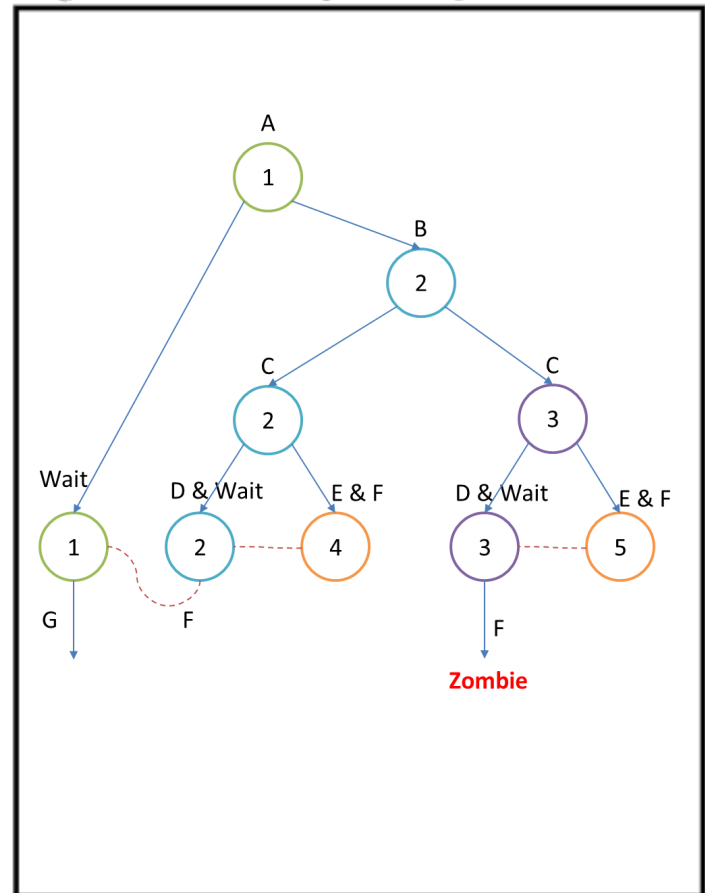
Linux/Unix Structure: 3rd One

Question 4 [11 Marks]

A. Draw the process tree for the following c code in the box. [5 Marks]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
```

```
int main() {
    printf("A\n");
    if (!fork()) {
        printf("B\n");
        fork();
        printf("C\n");
        if (fork()) {
            printf("D\n");
            int status;
            wait(&status);
        }
        else {
            printf("E\n");
        }
        printf("F\n");
    }
    else {
        wait(NULL);
        printf("G\n");
    }
    return 0;
}
```



B. Highlight all the Zombies in the tree created above. (Marking invalid zombies will result in deduction of 50% of obtained marks of Part A). [3 Marks]

C. Consider the following code: [3 Marks]

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/wait.h>
int main()
{
    printf("Start...\n");
    execl("./exe", ".exe", NULL);
    int id=fork( );
    if(id>0)
        printf("Parents\n");
    else
        printf("Child\n");
    printf("End,,,\n");
    return 0;
}
```

it is placed in file “Source.c” and compiled using the following command.

gcc -o exe Source.c

What is the output of given code?

Start...

Start...

Start...

.

.

.

GOOD LUCK 😊