National University of Computer and Emerging Sciences

FAST School of Computing **Spring 24** **Islamabad Campus**

CS-2006: Operating Systems

Monday, 26 February, 2024

Course Instructors

Dr. Faisal Cheema, Dr. Adnan Tariq, Miss. Maryam Shahbaz, Miss Rabail , Mr. M. Adil ur Rehman

Serial No:

Sessional 1

Total Time: 1 Hour

Total Marks: 60 Marks

_____

Signature of Invigilator

_____	_____	_____	_____
Student Name	Roll No.	Course Section	Student Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Answer sheet has been provided on th very LAST PAGE. It is mandatory to fill answer sheet. Otherwise,your exam won't be evaluated and you will be received zero.
2. Please refrain from removing the answer sheet of MCQs from exam. If you detect answer sheet from exam, your exam will be canceled.
3. Mark your roll number in the answer sheet. Use 6 digits of your registration number (ignoring alphabet) to fill roll number.
4. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
5. If you need more space write on the back side of the paper and clearly mark question and part number etc.
6. 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.
7. 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-4	Total
Marks Obtained				
Total Marks	15	15	30	60

Question 1 [15 marks]

Consider the following C Program named "fork-tree.c". Assume there are no errors in the program and everything works. Dry run in the rough space provided. Then answer the questions below.

```
//C program named as "fork-tree.c"
int main()
{
  for (int i = 0; i < 3; i++)
  {
      if (fork() == 0) {
            printf(" * ");
            printf("My process ID is %d\n", getpid());
            printf("My parent process ID is %d\n", getppid());
            exit(0);
      } //end if (fork() == 0)
      else
      {
            for (int j = 0; j < 2; j++) {
                  if (!fork() != 0) {
                        printf(" + ");
                        printf("My process ID is %d\n", getpid());
                        printf("My parent process ID is %d\n", getppid());
                        exit(0);
                   } //end if (!fork() != 0)
            } //end for j
            for (int k = 0; k < 2; k++)
                  wait(NULL);
      } //end else
   } //end for i
   printf(" # ");
   printf("My process ID is %d\n", getpid());
   printf("My parent process ID is %d\n", getppid());
}//end main()
```

Rough Work for Question 1

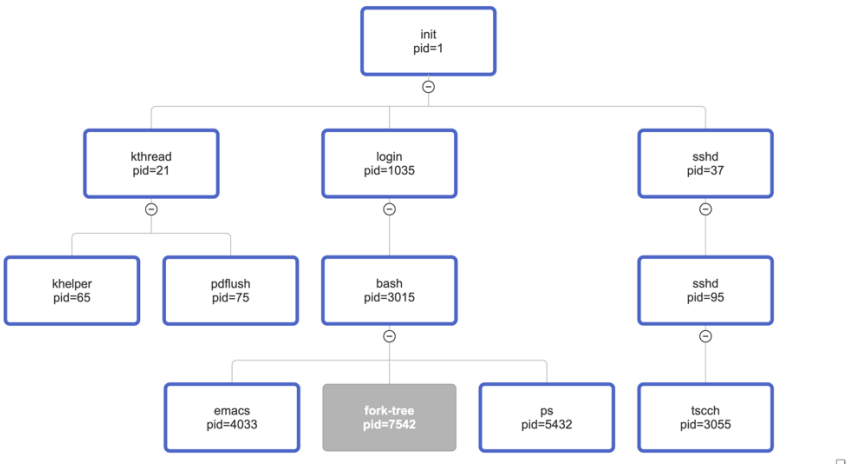
1. How many times " * " is displayed? [2 Mark]? Answer: _____3_____
2. How many times " + " is displayed? [2 Mark]? Answer: _____6_____
3. How many times " # " is displayed? [2 Mark]? Answer:_____1_____
4. How many maximum zombie processes can be created in the above code? [2 Mark]?

   Answer: _____9_____
5. Suppose the following lines are removed from the code:

```
for (int k=0;k<2;k++)
    wait( NULL) ;
```

   Now, if we re run again and fork tree parent process terminates as the very 1st process after completion. How many maximum processes can become orphan due to it? [2 Marks]

   Answer: _____all chILDREN_____
6. Suppose The fork-tree program has just initiated in the **new** state, and it has been assigned pid as **7542,** as shown below. Complete the Process Tree diagram for the fork-tree process below considering it runs successfully [5 marks]



All the created processes are direct children of the fork-tree process.

Question 2 [10 Marks]

Analyze the program by drawing process tree and dry run in the rough space provided. Then answer the questions provided in the table below. No answer will be accepted without the matching rough work according to these instructions.

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

```
int main() {
    pid_t pid1, pid2, pid3, pid4, pid5;
    pid1 = fork();
    if (pid1 == 0 || (pid2 = fork()) == 0) {
        printf("A\n");
        exit(0);
    } else {
        wait(NULL);
        pid3 = fork();
        if (pid3 == 0 && (pid4 = fork()) == 0) {
            printf("B\n");
            exit(0);
        } else if (pid3 > 0) {
               int s;
            waitpid(pid3, &s, 0);
               printf("Status of PID3 = %d\n", s);
            if ((pid5 = fork()) == 0) {
                printf("C\n");
                exit(0);
            } else {
                     wait(NULL);
                printf("D \n");
                     return 0;
            }  } }
    return 1;}
```

Question	Answer
Total number of Processes created including the main process [6 Marks]	7
Number of Zombie Processes [3 Marks]	3
How many times B will be printed? [2 Marks]	1
Consider removing the last code statement "return 1" from the code snippet provided above, What would be changed in the output? [2 Marks]	Status of PID=Garbage (maybe 0) or main process would be waiting indfinitely
Consider removing the last code statement "return 1" from the code snippet provided above, Number of Zombie Processes [2 Marks]	4 or 3

,

Rough Work for Question 2

Question 3 [Multiple choice questions]

1. Which of the following statements about interrupts is true?

A. Interrupt vector tables are used to store addresses of I/O devices only.
B. Interrupts with I/O involve the processor waiting indefinitely for device responses.
C. Interrupts with Direct Memory Access (DMA) bypass the processor entirely.
D. Interrupts cannot occur during system calls.
E. None of the above.

2. Which of the following statements best describes a limitation of the monolithic kernel architecture?

A. It provides better isolation between kernel components compared to microkernel architectures.
B. It allows for easy extension and modification of kernel functionality without recompilation.
C. It typically results in lower context-switching overhead compared to microkernel architectures.
D. It facilitates rapid development of new device drivers and file systems due to its highly modular design.
E. It tends to have a larger memory footprint due to the inclusion of all operating system services in a single address space.

3. **Which of the following statements best describes a key advantage of the microkernel architecture?**

A. It offers superior performance in I/O-bound workloads compared to monolithic kernel architectures.
B. It allows for seamless integration of third-party device drivers and file systems without requiring modifications to the kernel.
C. It enables better real-time responsiveness due to reduced context-switching overhead.
D. It enhances system security by minimizing the attack surface exposed by the kernel.
E. It simplifies debugging and maintenance by consolidating all operating system services within the kernel space.

4. 4. Which of the following statements best illustrates a fundamental difference between Docker containers and Virtual Machines (VMs)?

A. VMs require a hypervisor to manage hardware virtualization, while Docker containers leverage operating system-level virtualization.
B. Docker containers provide complete isolation of resources, including CPU, memory, and storage, similar to VMs.
C. VMs offer superior performance and efficiency compared to Docker containers due to their lightweight architecture.
D. Docker containers share the host operating system kernel, enabling faster startup times and reduced resource overhead.
E. Docker containers are immutable and stateless, whereas VMs support dynamic reconfiguration and state persistence.

5. Which of the following does NOT cause a trap?

A. user program divides a number by zero.
B. The operating system kernel executes a privileged instruction.
C. A programmable interval timer reaches its specified time.
D. A user program executes an interrupt instruction.
E. None of the above.

6. Which of the following components is MOST LIKELY to always be part of the microkernel, even in a highly security-focused design?

A. Process scheduling and context switching
B. Inter-process communication (IPC) mechanisms
C. Memory management and protection mechanisms
D. System call interface and handling
E. Basic timer and interrupt handling

7. The system receives an interrupt from the disk controller indicating the I/O operation for process A has finished. Which of the following is most likely to occur first?

A. Process A's interrupt service routine (ISR) is immediately executed.
B. The operating system saves the CPU context and schedules another process.

C. The operating system updates A's PCB and places it in the Ready state.
D. The I/O completion notification is sent directly to process A's code.
E. The interrupt is ignored until process A requests the I/O data.

8. Consider a multi-user system implementing five state process model, currently having five processes: A, B, C, D, and E. The current state of the processes is:

A. Process A is in the Running state
B. Process B is waiting for an I/O operation (Blocked state).
C. Process C is ready to run (Ready state).
D. Process D is newly submitted and needs resources allocated (New state).
E. Process E is finished execution and waiting for its resources to be reclaimed (Terminated state).

9. In a Process Control Block (PCB), which of the following data elements is LEAST LIKELY to be found?

A. Open Files Table
B. Program Counter (PC)
C. Register Values
D. Complete source code
E. Memory Layout information (e.g., stack base, heap limits)

10. The system receives an interrupt from the disk controller indicating the I/O operation for process A has finished. Which of the following is most likely to occur first?

A. Process A's interrupt service routine (ISR) is immediat executed.
B. The operating system saves the CPU context and schedules another process.
C. The operating system updates A's PCB and places it in the Ready state.
D. The I/O completion notification is sent directly to process A's code.
E. The interrupt is ignored until process A requests the I/O data.

11. 11. Which of the following interrupt types most likely corresponds to each event?

A. Event 1: Trap, Event 2: System call, Event 3: Abort
B. Event 1: Fault, Event 2: Interrupt, Event 3: Abort
C. Event 1: Abort, Event 2: System call, Event 3: Trap
D. Event 1: Exception, Event 2: Trap, Event 3: Fault
E. The provided information is insufficient to determine the exact types.

12. A process stack does not contain:

A. Function parameters
B. Local variables
C. Return addresses
D. PID of child process
E. PID of itself

13. How is modularity added to the Linux kernel that is typically a monolithic kernel:

A. By adding more system calls
B. Through loadable modules
C. By adopting micro-kernel approach in Linux
D. Both a and b

14. In the five-state model why would a process move from Running to Ready state?

A. The process has terminated
B. The process needs to perform an I/O operation
C. The process' time quantum has expired
D. The process needs to execute a system call

15. 15. A zombie process is a process that has terminated and

A. Its parent process has also terminated without applying the wait system call
B. The parent process is currently executing without applying the wait system call
C. The parent process is currently executing without applying the exit system call
D. It was created without any parent
E. None of these

16. Multiprogramming is:

A. An executable program that is composed of modules built using different programming languages.
B. Keeping several programs in memory at once and switching between them.
C. Having multiple processors execute different programs at the same time.
D. When a program has multiple threads that run concurrently

17. A process control block is:

A. A structure that stores information about a single process.
B. The kernel's structure for keeping track of all the processes in the system.
C. A linked list of blocked processes (those waiting on some event).
D. A kernel interface for controlling processes (creating, deleting, suspending).

18. Which state transition is not valid?

A. Ready → Blocked
B. Running → Ready
C. Ready → Running
D. Running → Blocked

19. Preemption is when an operating system moves a process between these states:

A. Running → Blocked
B. Running → Ready
C. Ready → Blocked
D. Blocked → Running

20. With DMA (Direct Memory Access):

A. The processor can read or write directly to a device.
B. The kernel can read or write directly to a process' memory without intermediate buffers.
C. A process can read or write to kernel memory without intermediate buffers.
D. The device can read or write directly to the system's memory

21. When a process is first launched, the operating system does not know the size of this segment:

A. text
B. data
C. bss
D. heap

22. Which of the option arranges the following technologies in the order from fastest to slowest:

A. Hard-disk drives, main memory, cache, registers
B. Registers, main memory, hard-disk drives, cache
C. Registers, cache, main memory, hard-disk drives
D. Cache, registers, main memory, hard-disk drives

23. You are a software developer working on the next generation of a popular smartphone operating system. The phone needs to be powerful, secure, and energy-efficient. Which of the following factors is most likely to favor a microkernel architecture for this smartphone operating system?

A. The need for extensive customization of the operating system for specific user needs.
B. The requirement for real-time performance for latency-sensitive applications like gaming.
C. The desire to leverage the latest hardware features for maximum processing power.
D. The need for strong security features to protect user data and privacy.
E. The goal of achieving the longest possible battery life for the smartphone.

24. 24. Your program forks a child process to perform a long-running task (i.e., longer than parent). However, you forget to wait for the child process to terminate and continue with your program's execution. What would happen to the child process?

A. Running normally, unaffected by the parent process.
B. Terminated immediately as it cannot be executed without a parent

C. Becomes an orphan process, detached from the parent but still running.
D. Enters a zombie state, lingering until explicitly reaped.
E. None of the options seems to be valid in this case.

25. A device driver runs on the microcontroller that is part of the external device

A. True
B. False
C. Depends on the device architecture

26. When interacting with a fast device, it can be better to spin wait than to use interrupts.

A. True
B. False
C. I/O devices cannot be fast

27. 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

28. 28. 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

29. 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

30. **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

MCQS SOLUTION

1	C	16	B
2	E	17	A
3	D	18	A
4	A	19	B
5	B	20	D
6	C	21	D
7	C	22	C
8	E	23	D
9	D	24	C
10	C	25	B
11	B	26	A
12	D	27	C
13	B	28	B
14	C	29	C
15	C	30	A