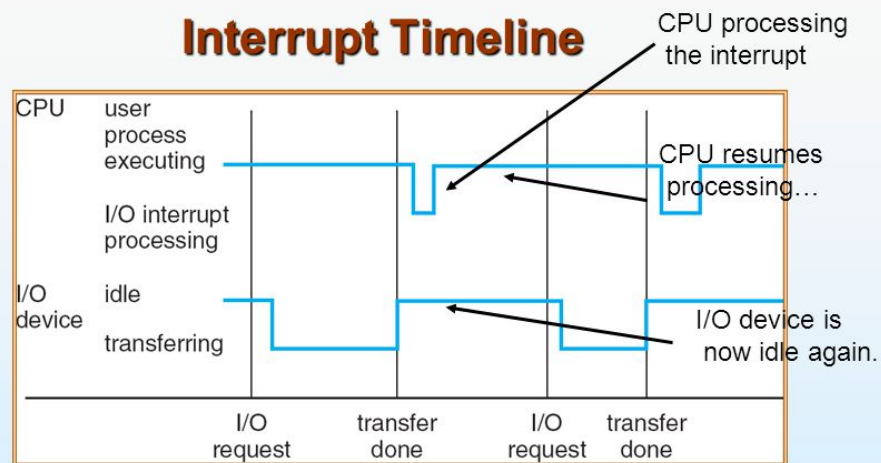


5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



Interrupt Timeline



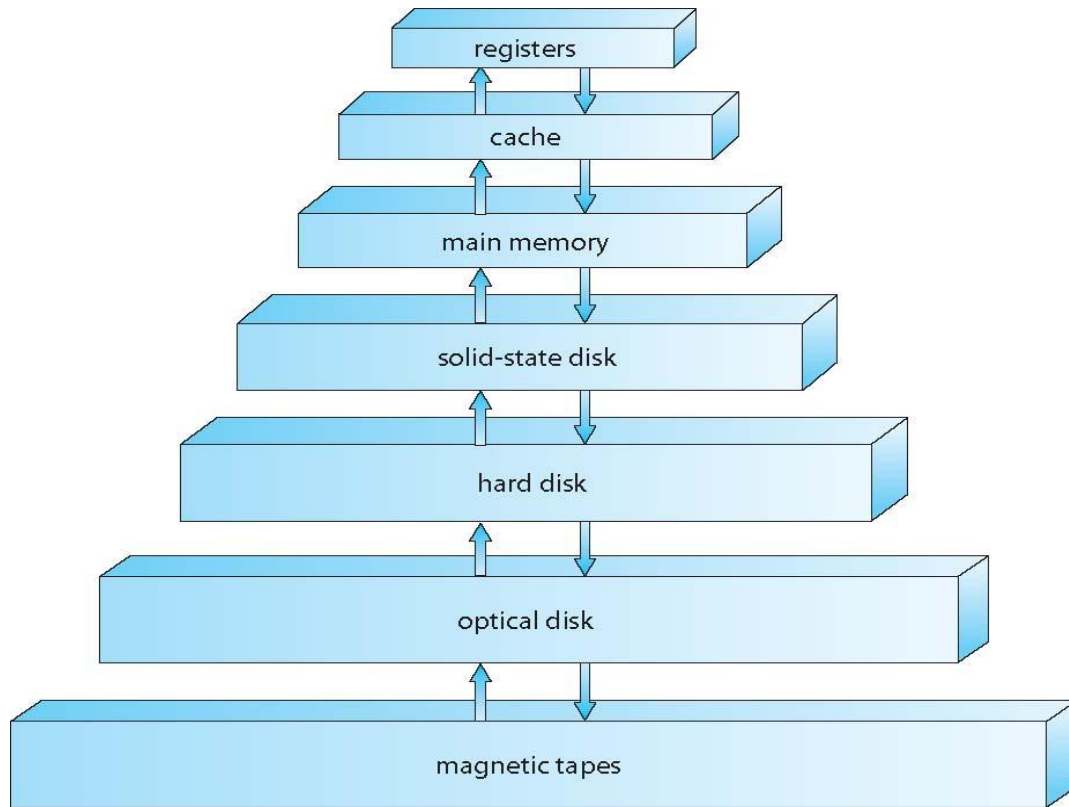
Consider: the CPU is busy executing some process.
In the background, an I/O device, say a disk, is transferring data from itself into primary memory.
When the transfer is complete, a signal (interrupt) is sent to the CPU.
The CPU can suspend 'current' operations, and process the I/O interrupt
Of course at this time, the I/O device becomes idle
Note the CPU time to process the interrupt is very small (much smaller than chart indicates).
Then the CPU resumes normal processing (whatever that might be)
It might resume suspended process or start executing a different process.

This is sufficient for now. Much more later.



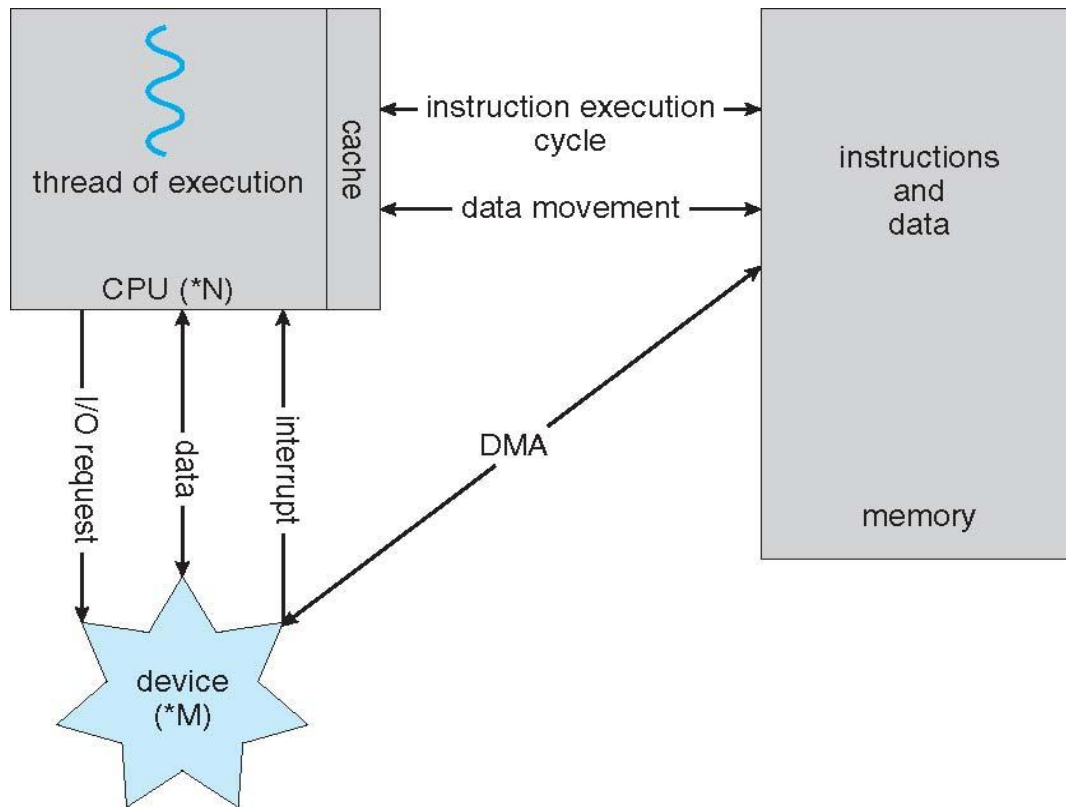
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



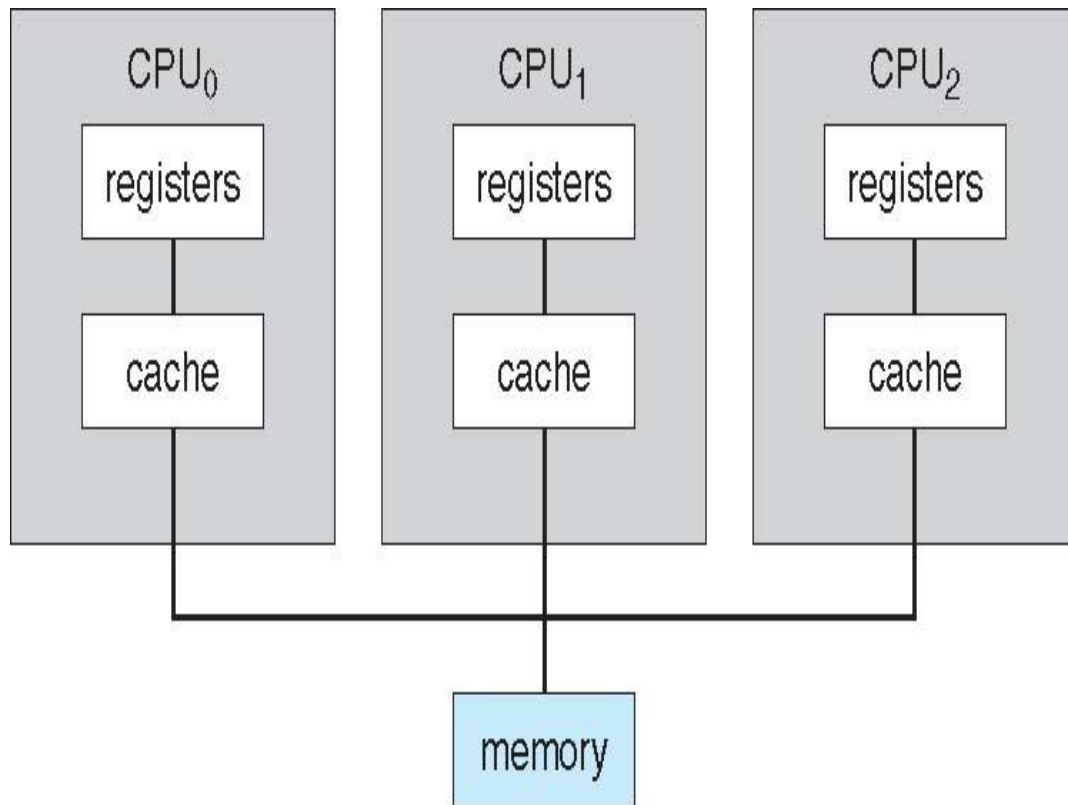
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



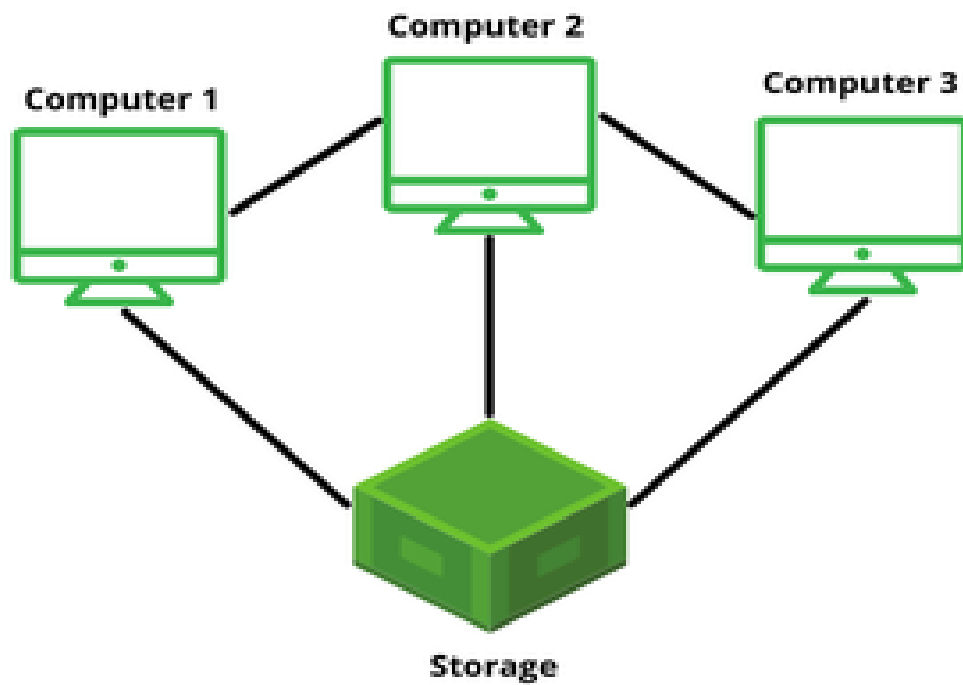
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



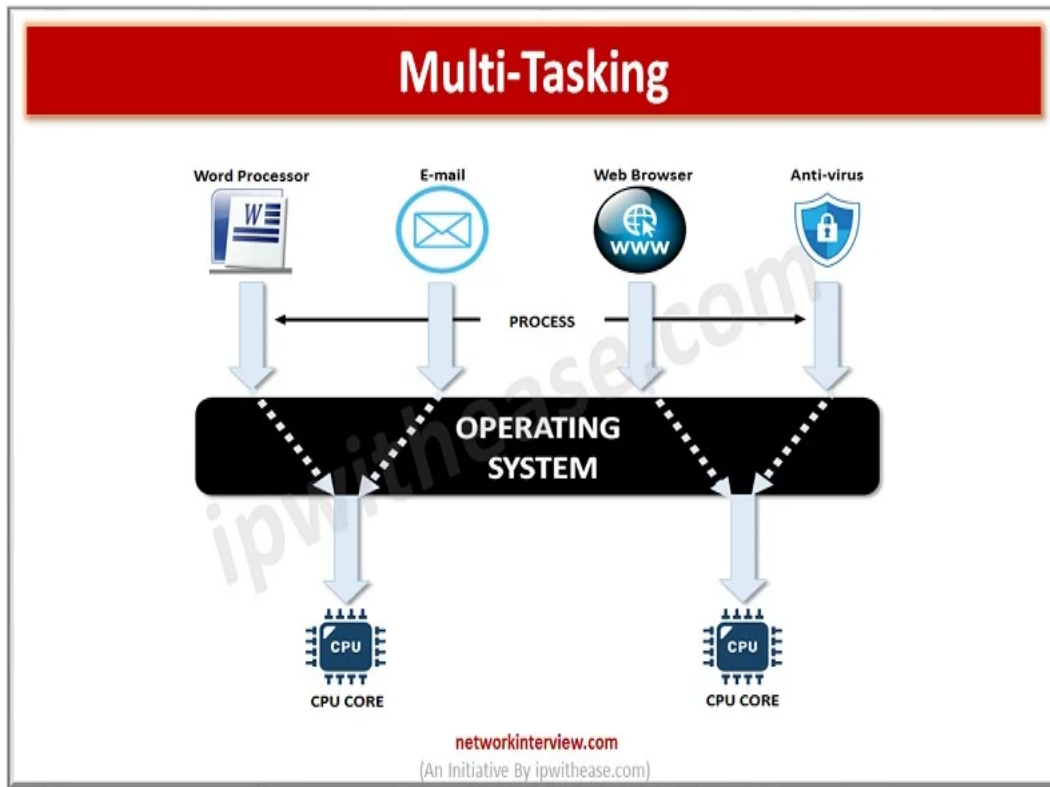
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



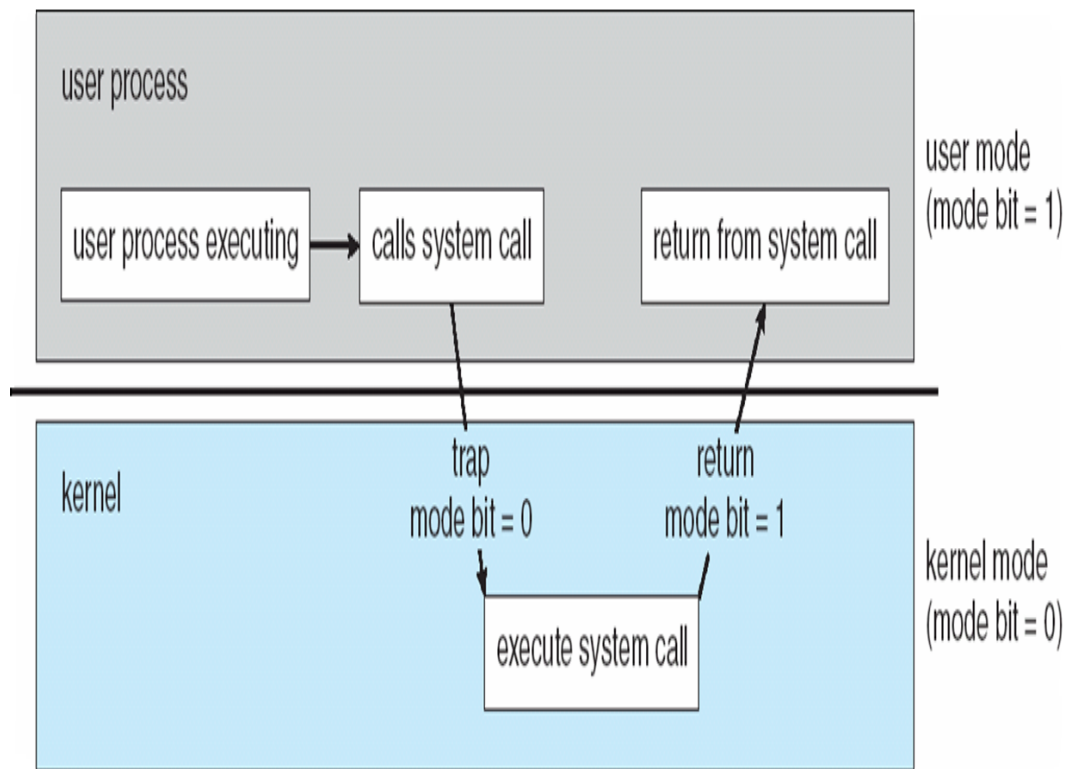
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



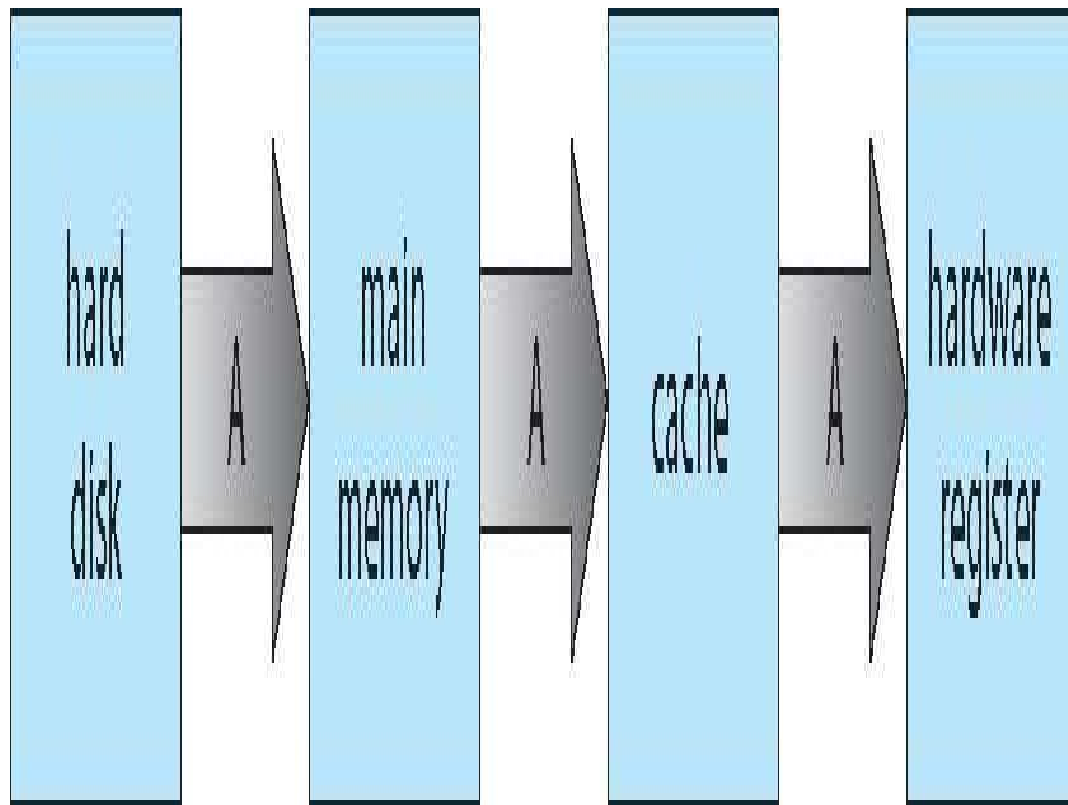
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



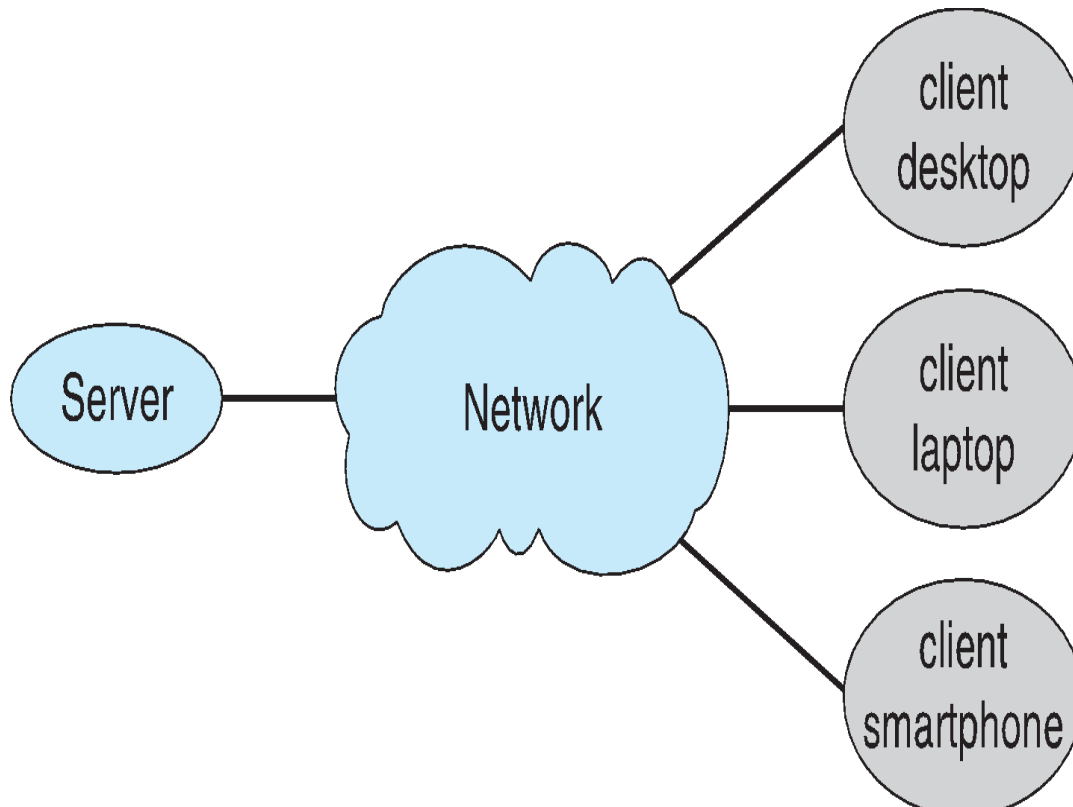
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



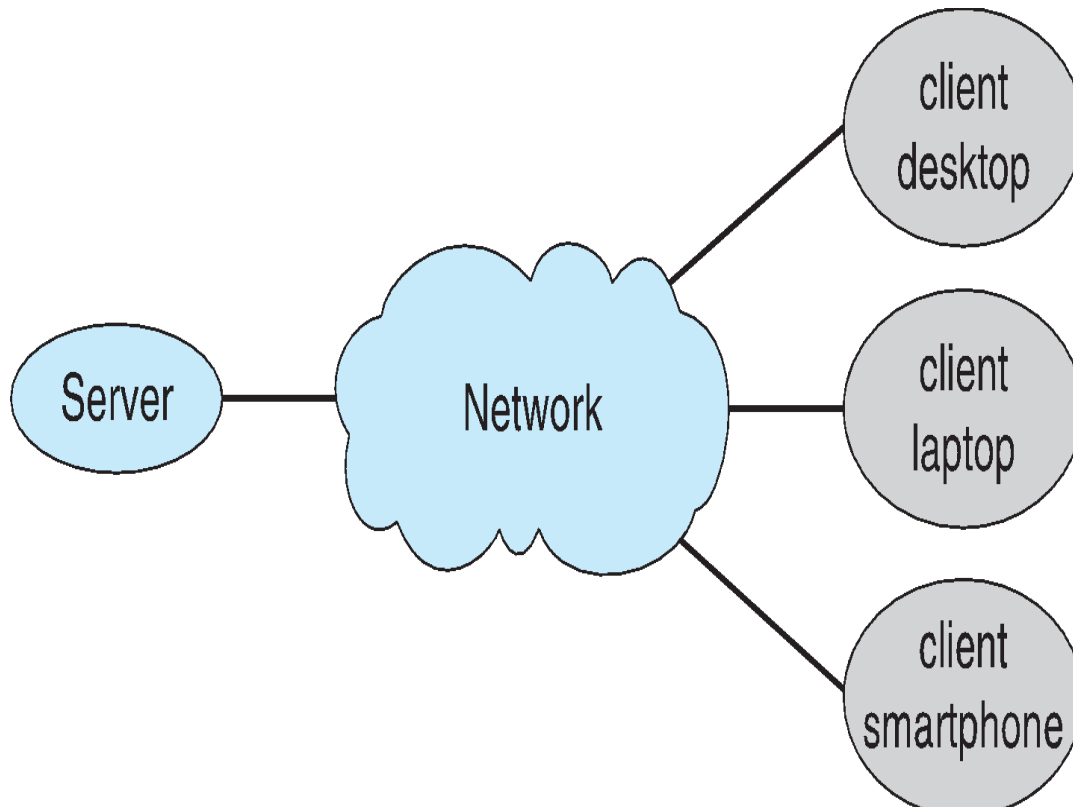
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



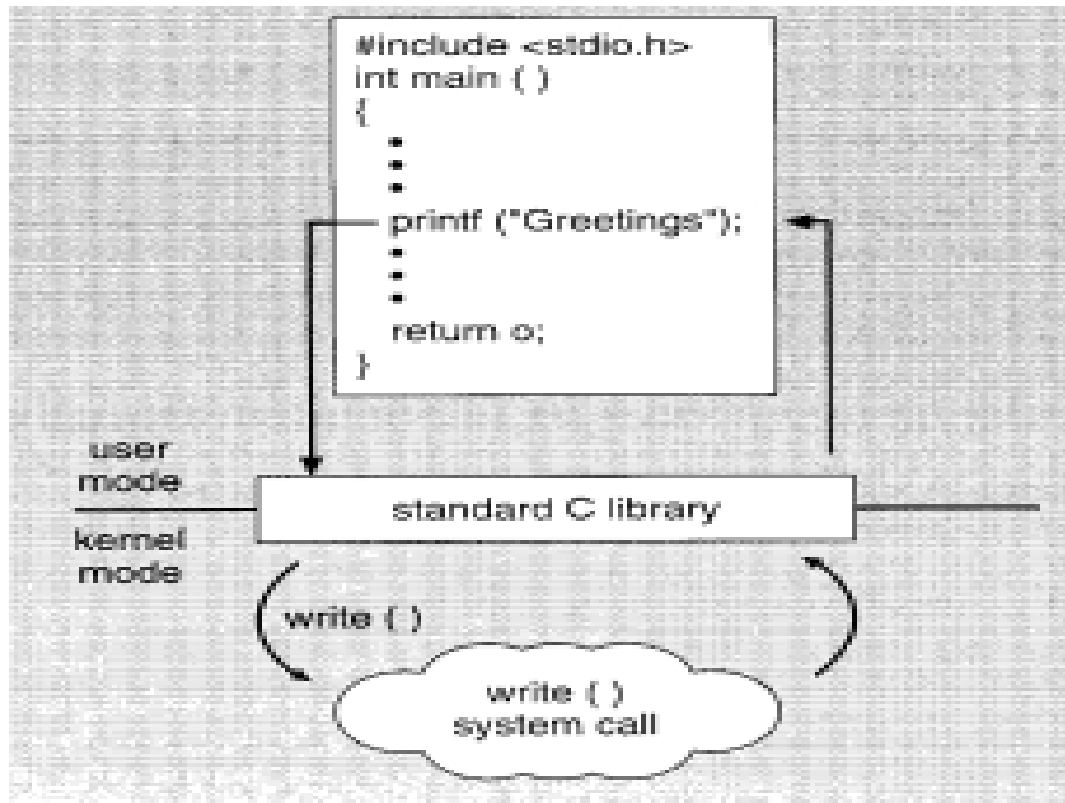
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



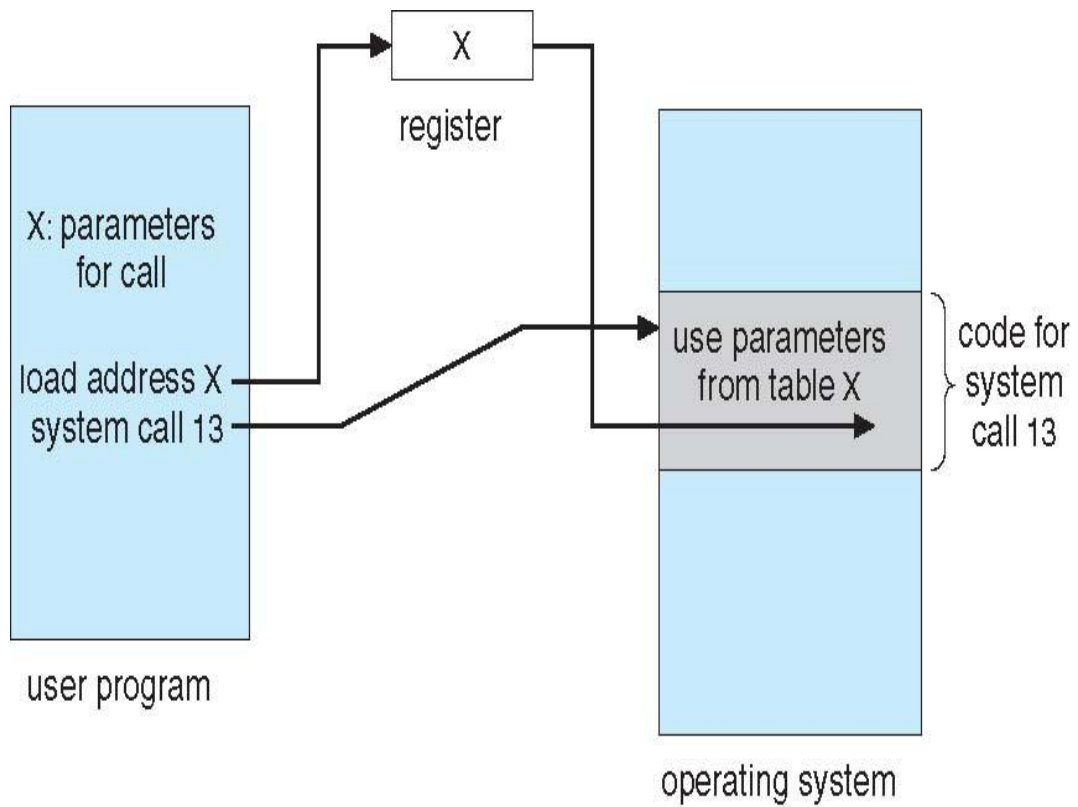
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



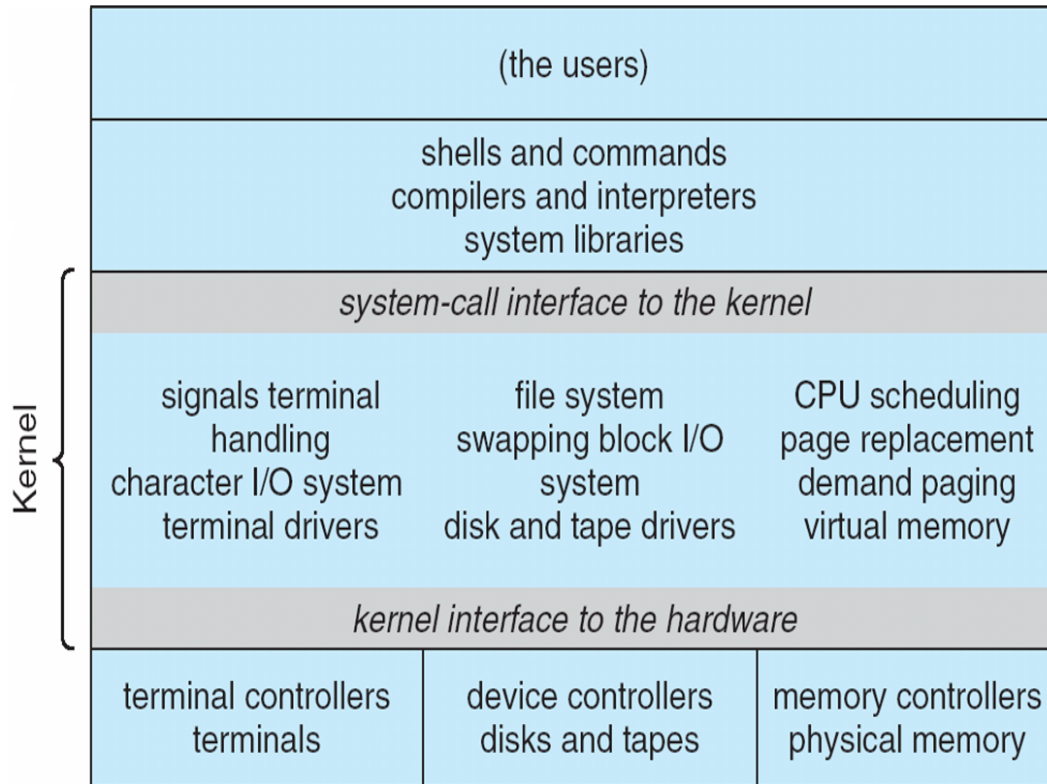
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



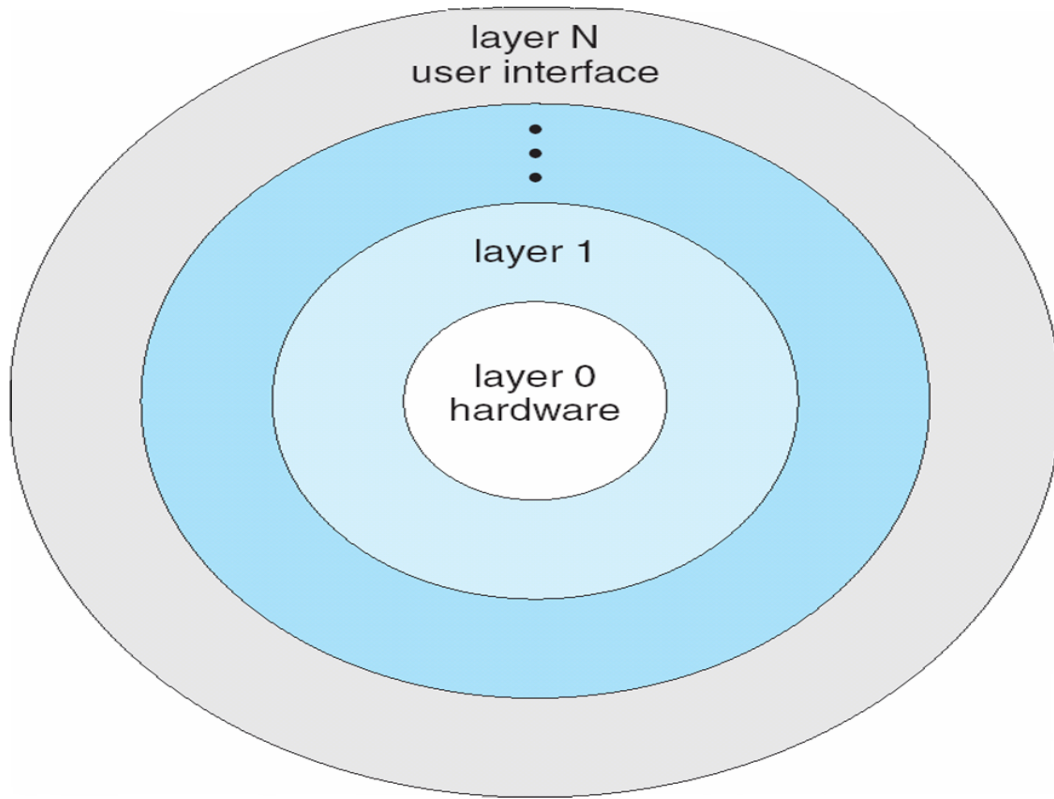
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



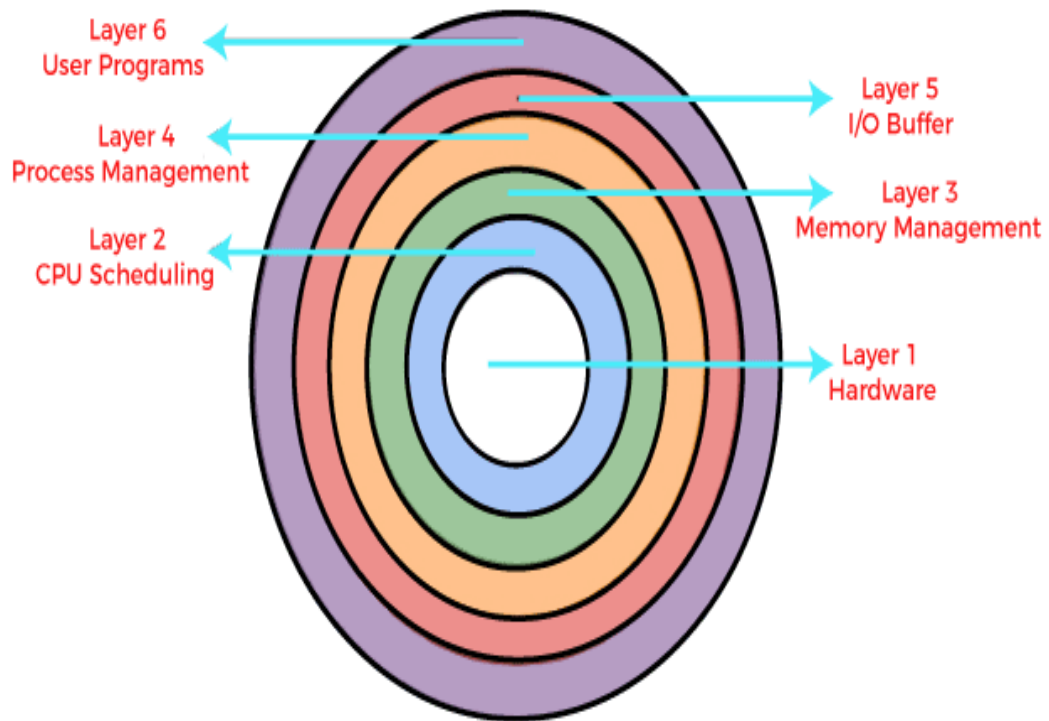
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



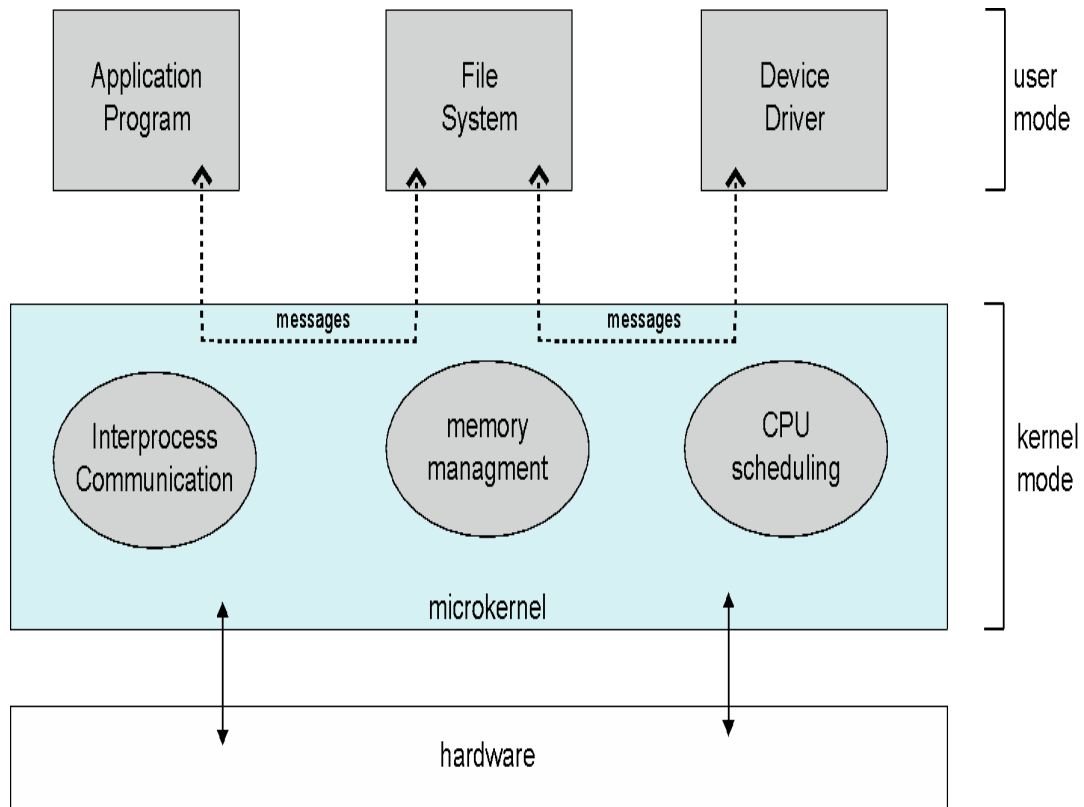
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



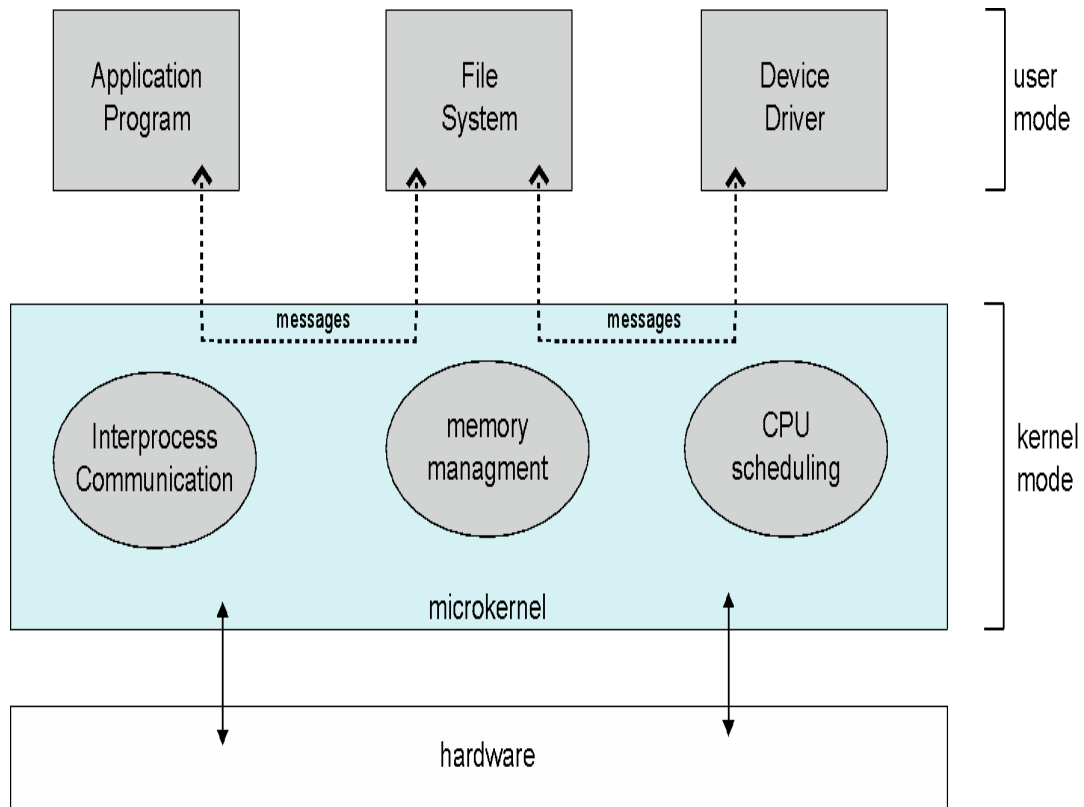
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



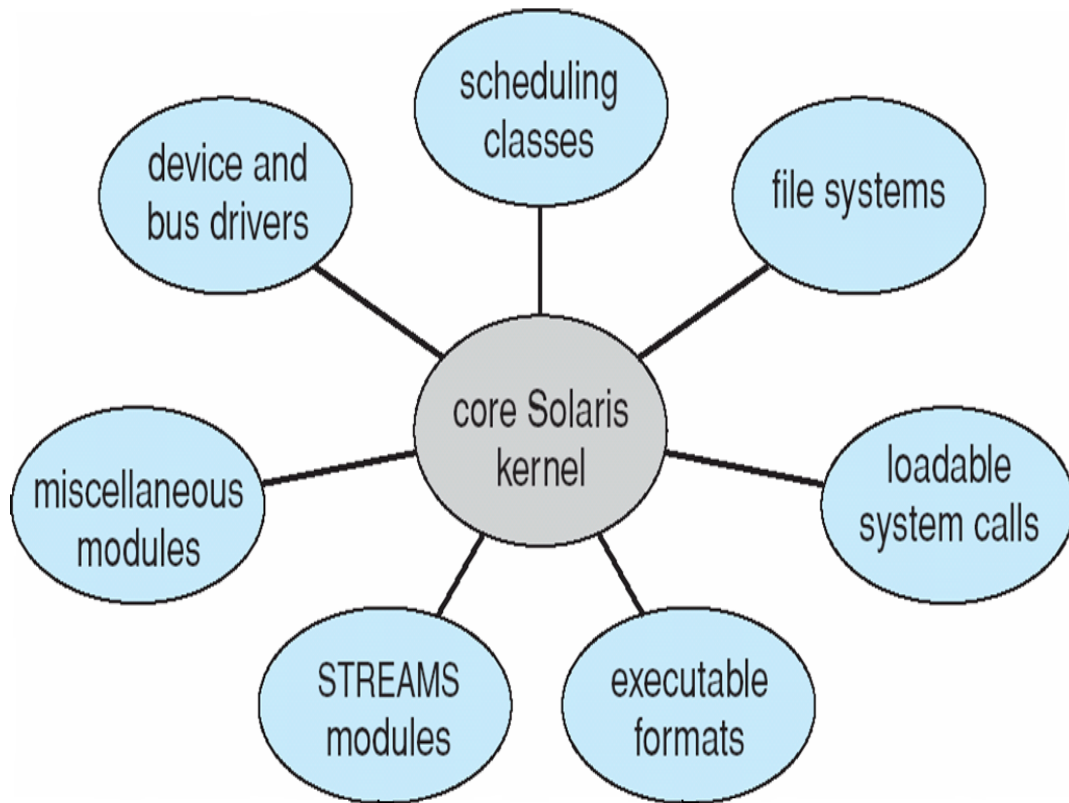
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



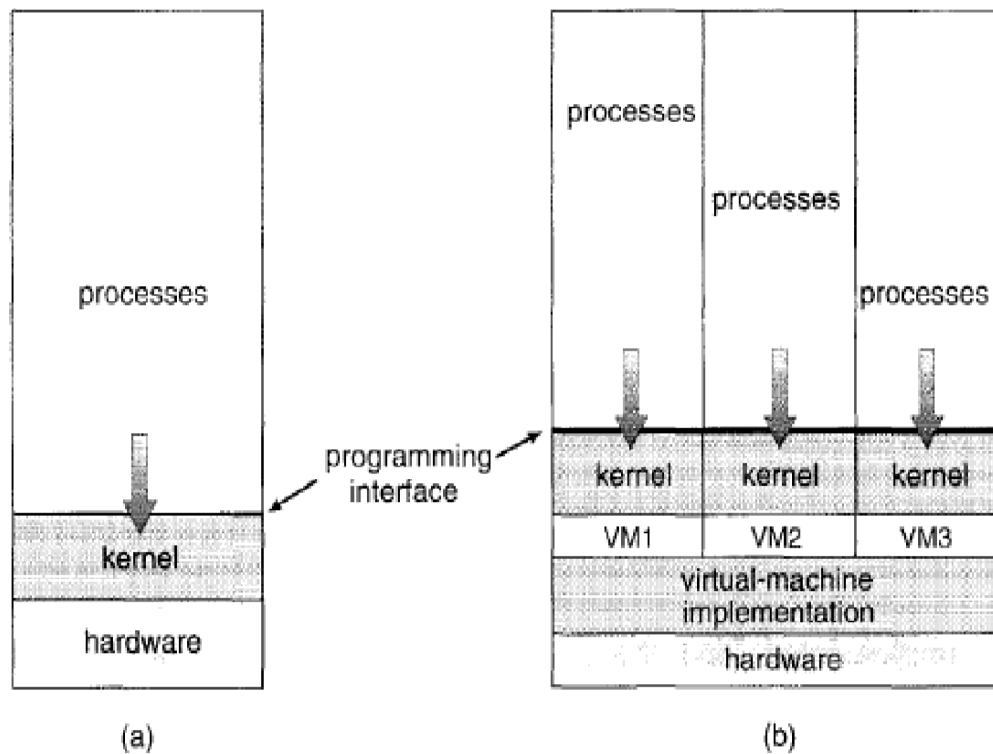
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



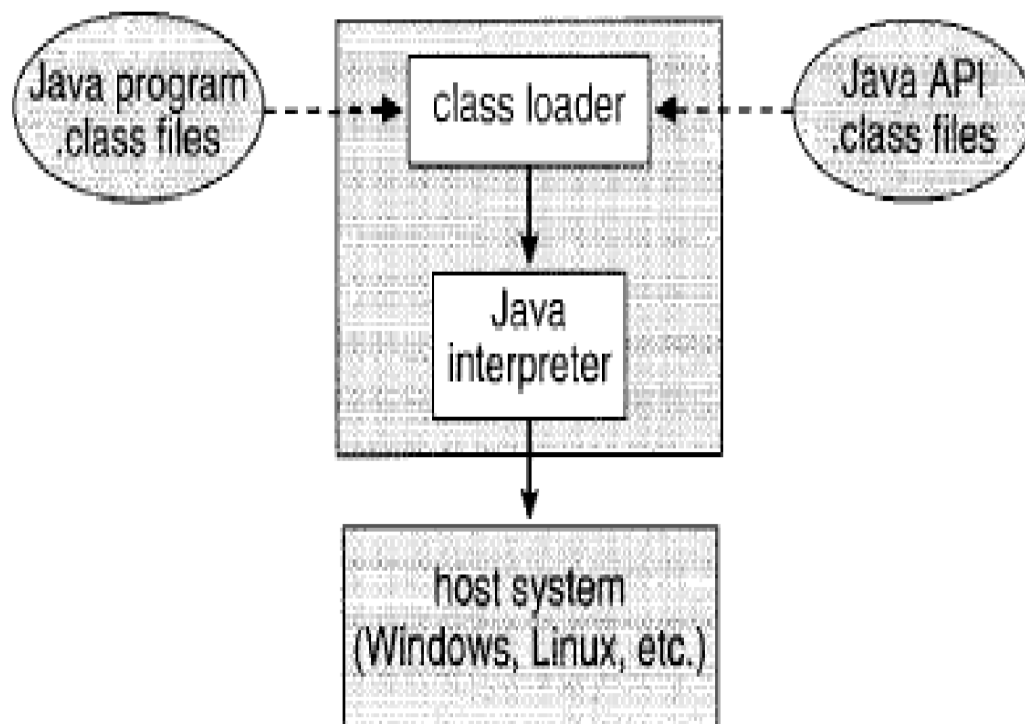
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



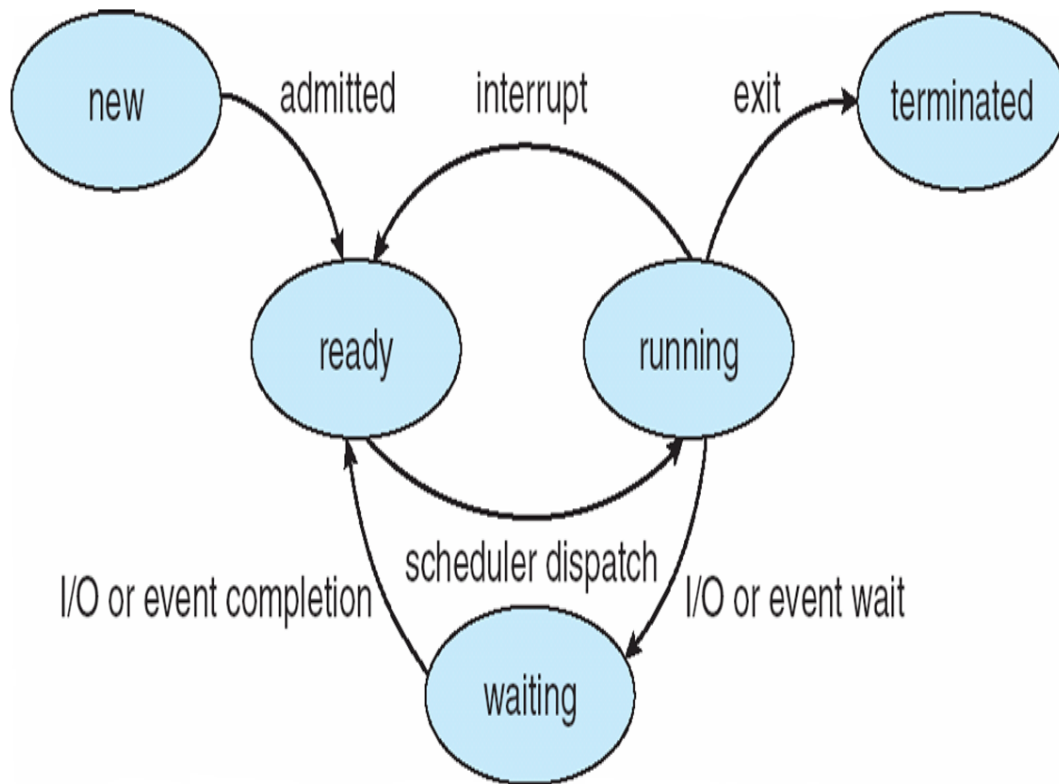
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



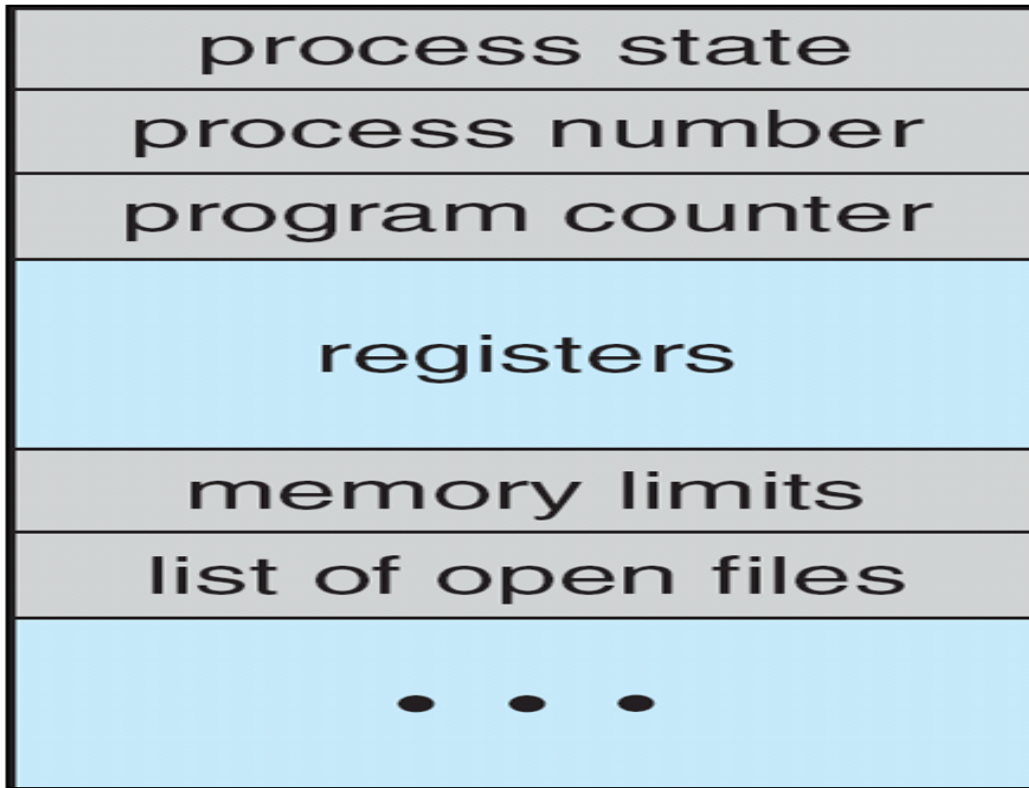
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



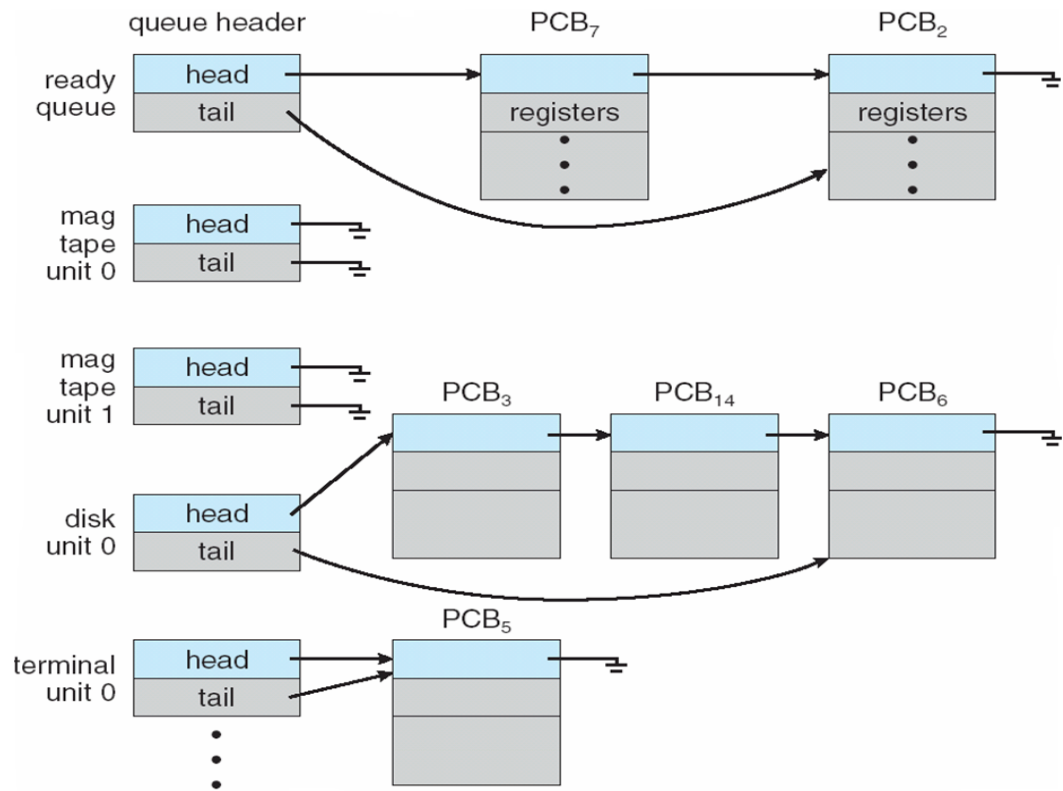
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.


```

#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid;

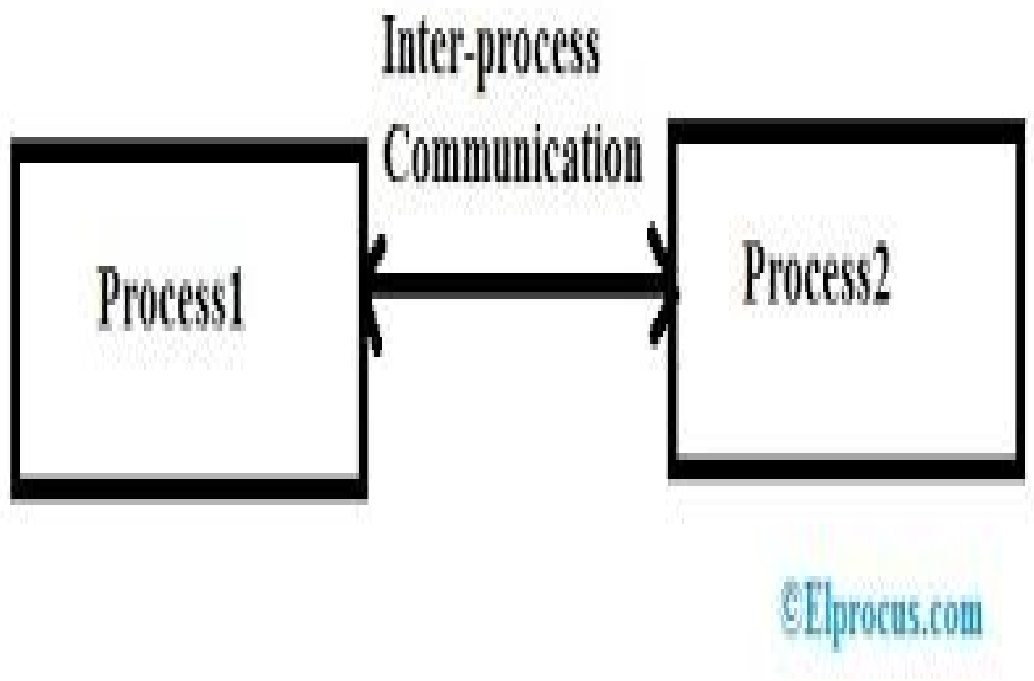
    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        exit(-1);
    }
    else if (pid == 0) { /* child process */
        execlp("/bin/ls", "ls", NULL);
    }
    else { /* parent process */
        /* parent will wait for the child to complete */
        wait(NULL);
        printf("Child Complete");
        exit(0);
    }
}

```

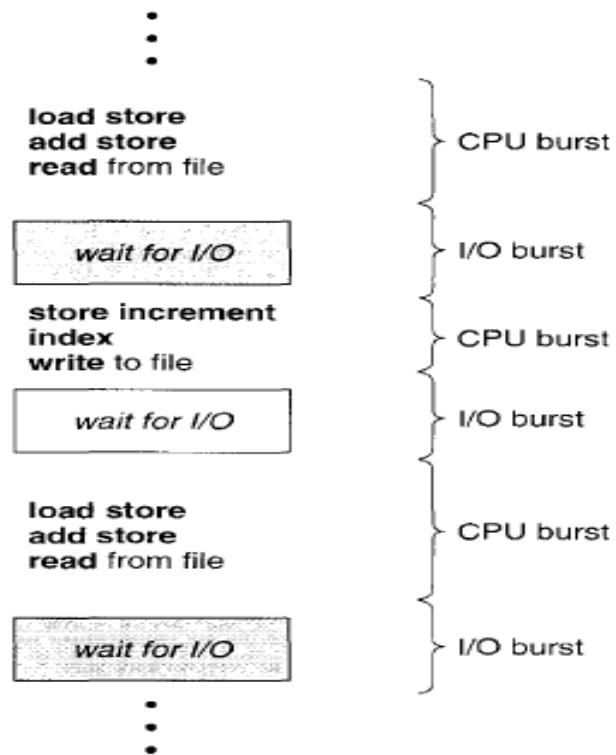
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



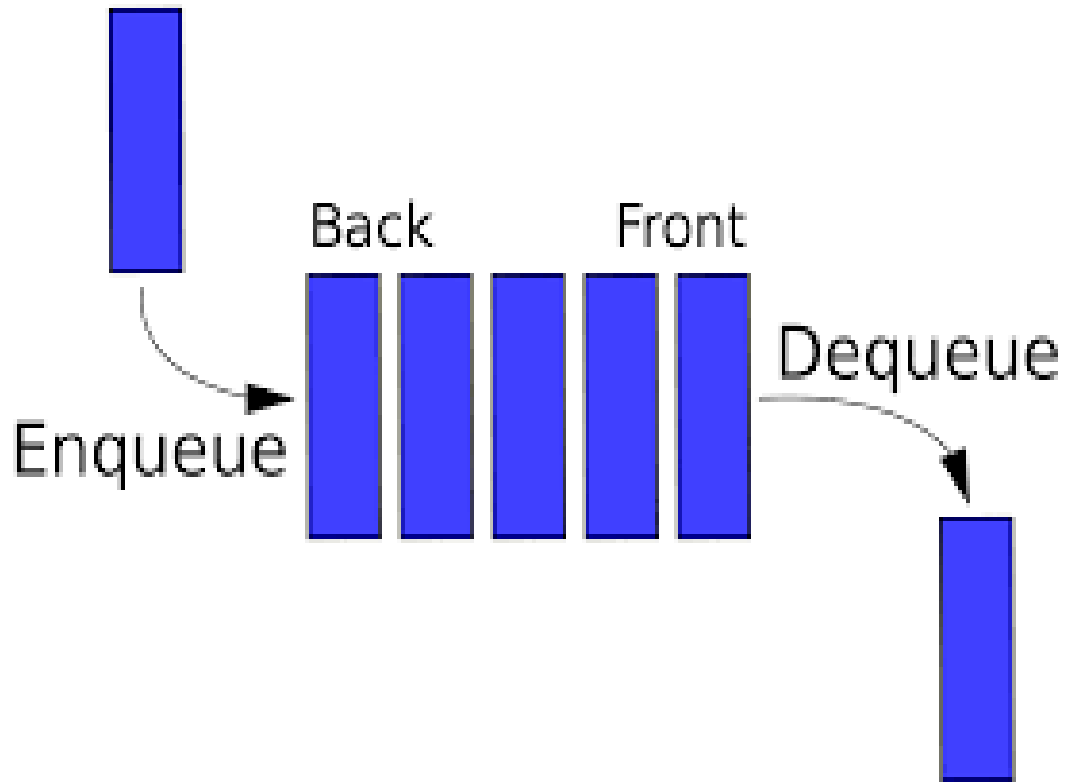
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



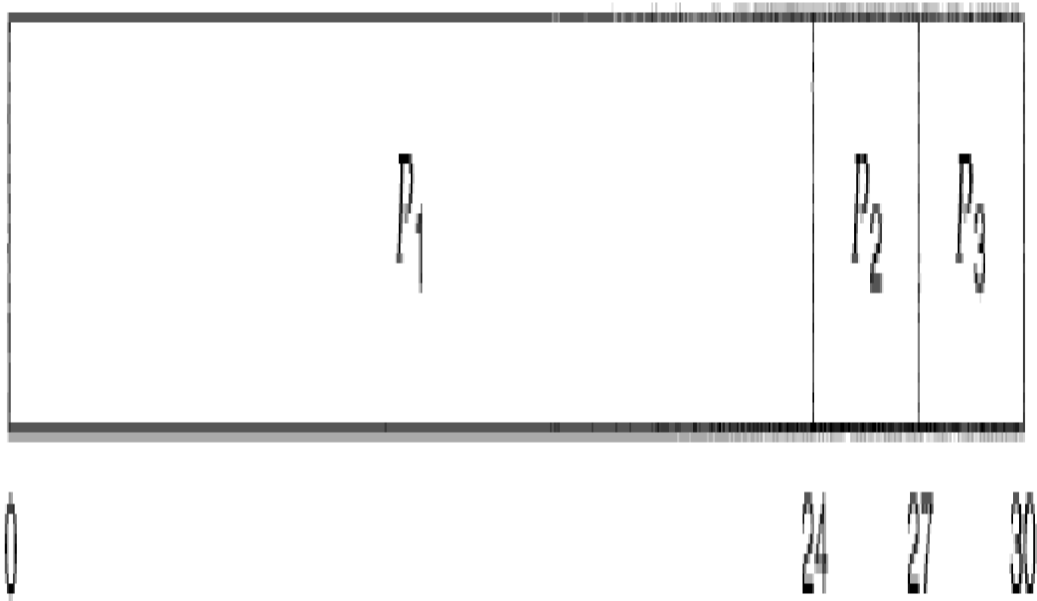
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



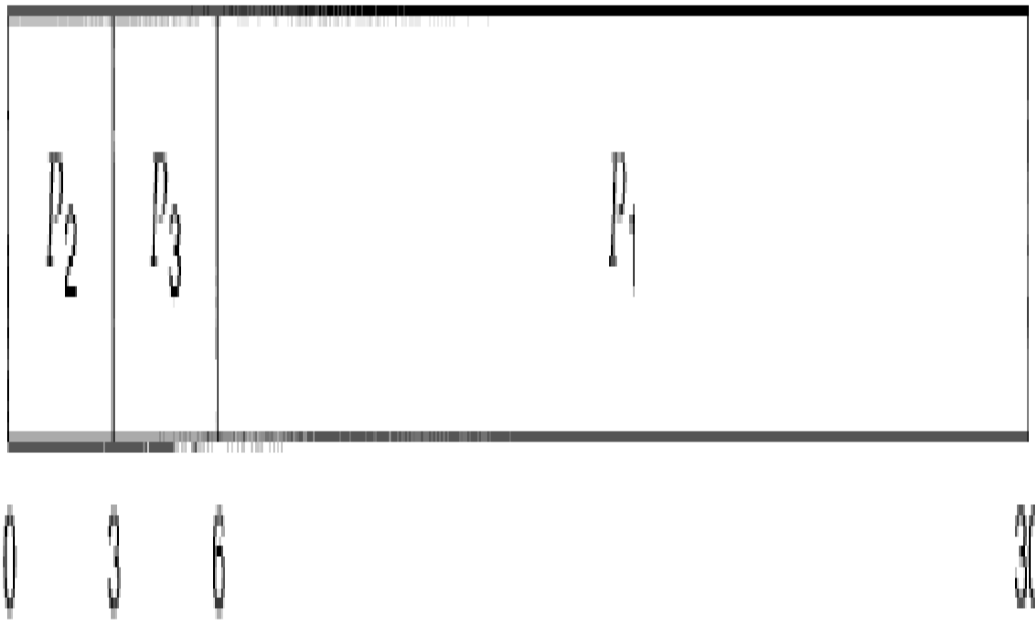
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



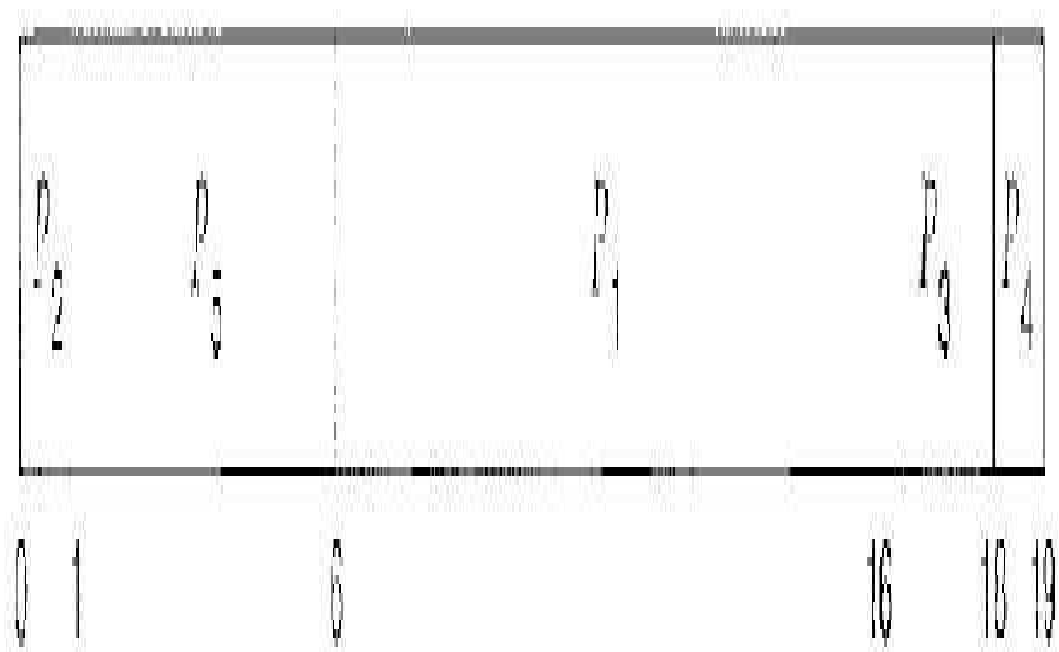
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

Process	Burst Time	Arrival Time
P1	16	0
P2	5	1
P3	6	2
P4	1	3
P5	2	4

5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



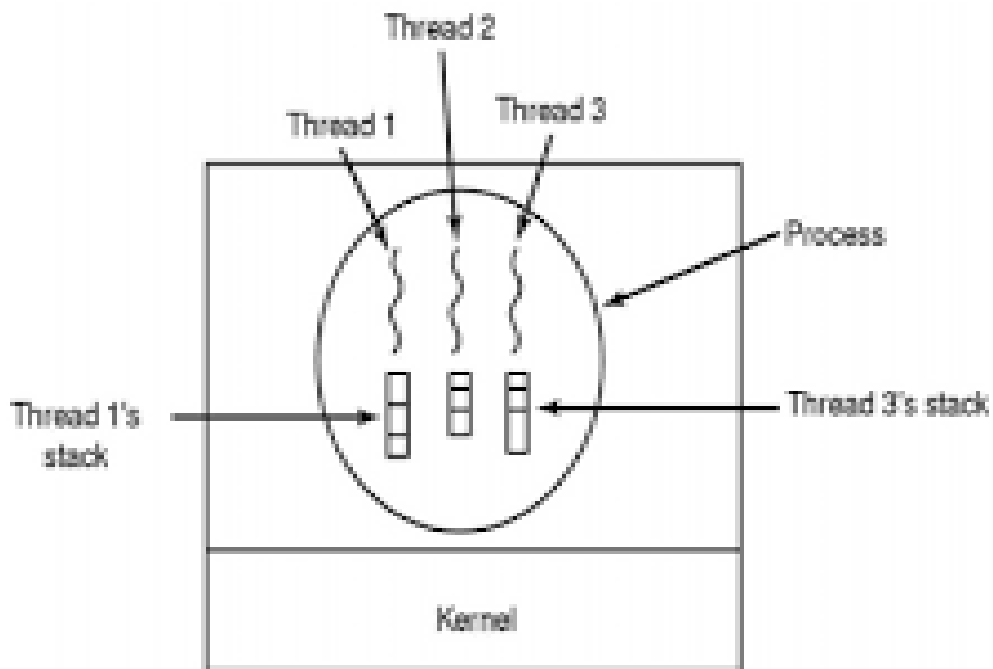
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

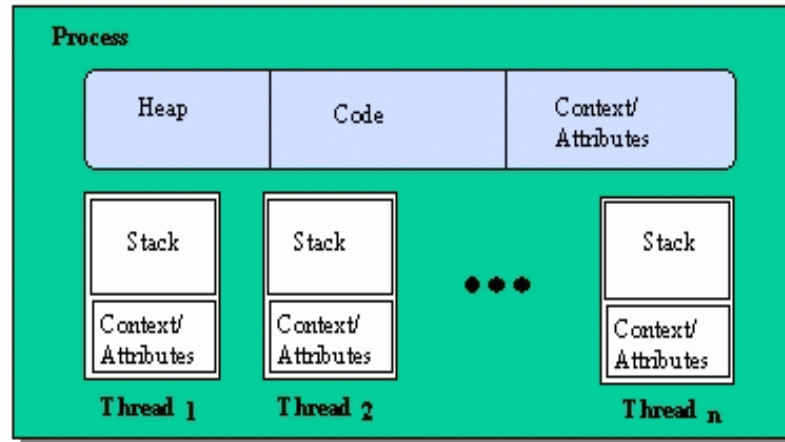


Each thread has its own stack.

5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

Threads and Operating System Processes



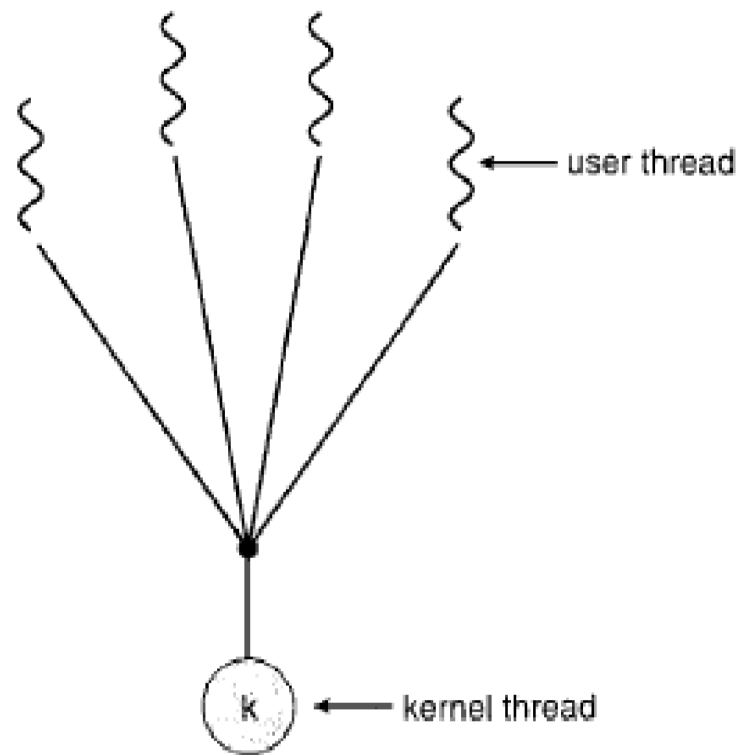
An Operating system process provides a protected address space.
Many threads may execute within the address space.
Each thread has its own stack & context (saved registers).

Concurrent programming- SE II

19

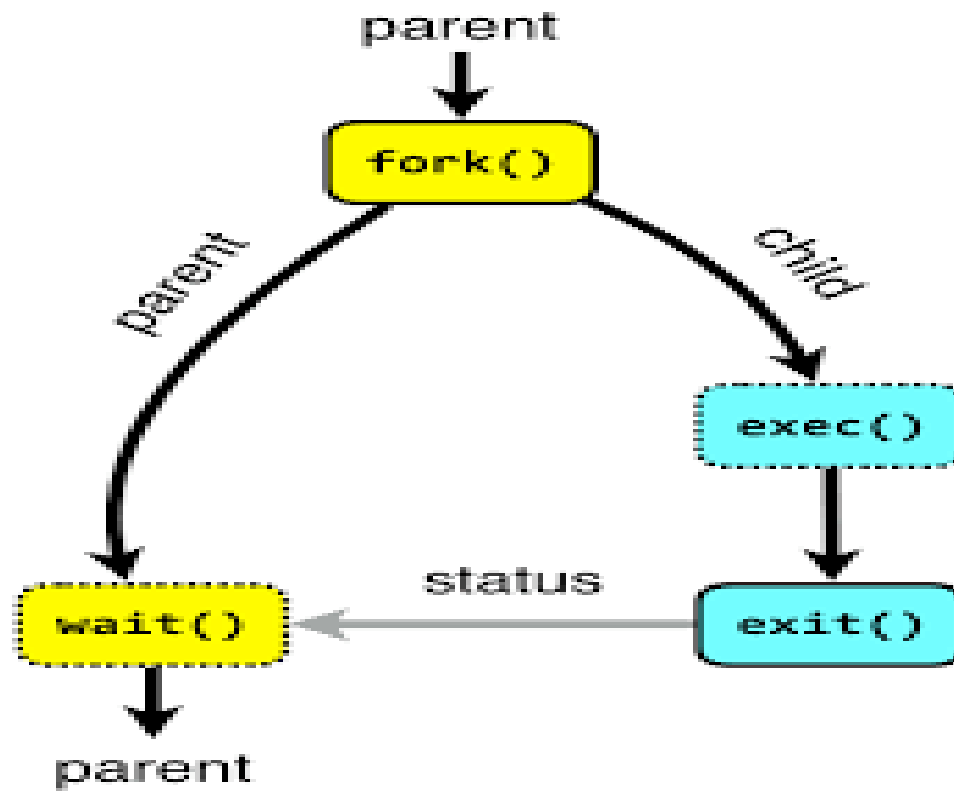
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



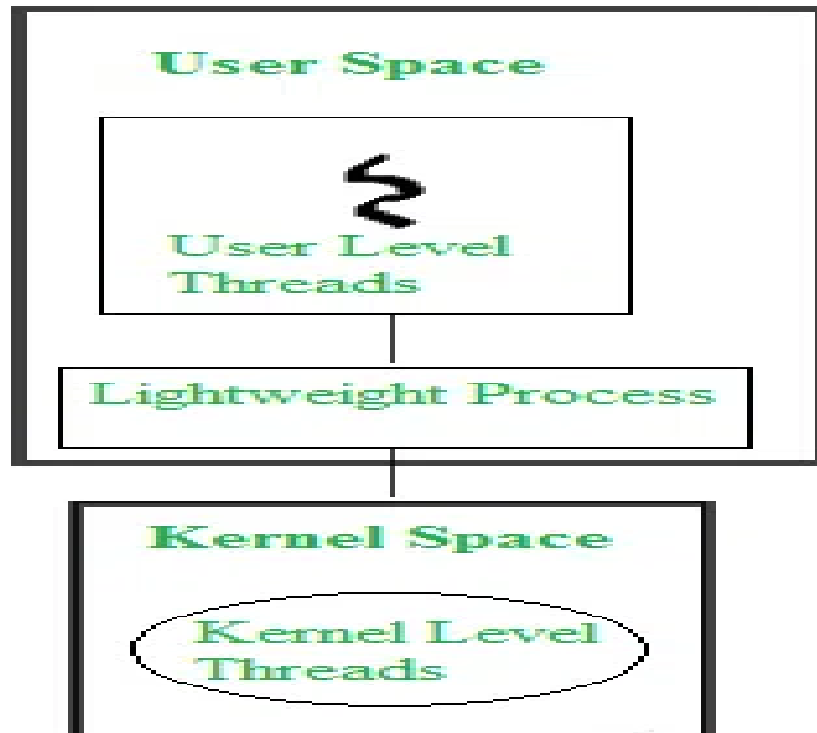
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

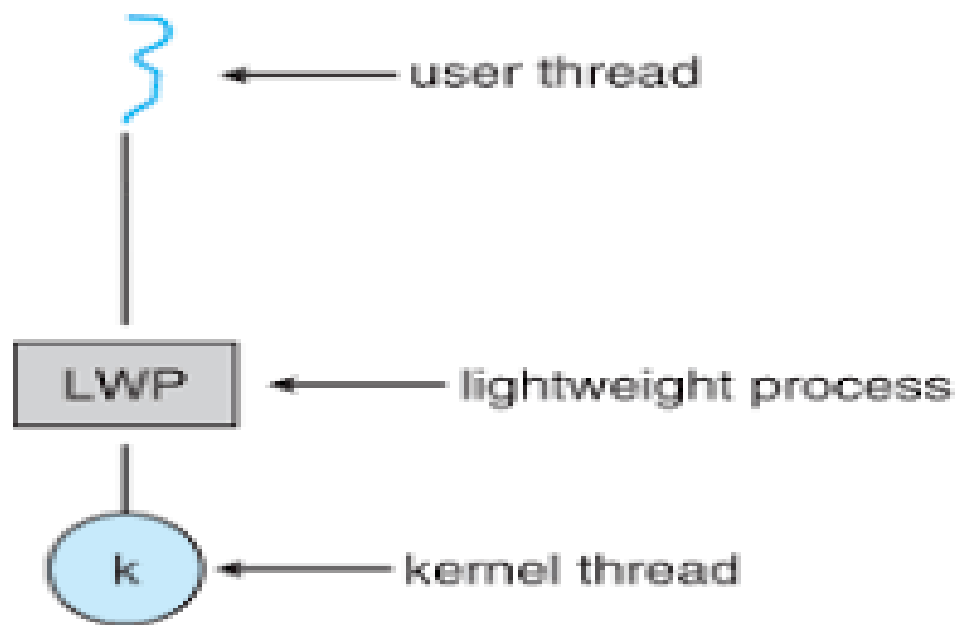


Figure 4.13 Lightweight process (LWP).

5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

do {

entry section

critical section

exit section

remainder section

} while (TRUE);

5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

do {

```
flag[i] = TRUE;  
turn = j;  
while (flag[j] && turn == j);
```

critical section

```
flag[i] = FALSE;
```

remainder section

} while (TRUE);

5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

do {

acquire lock

critical section

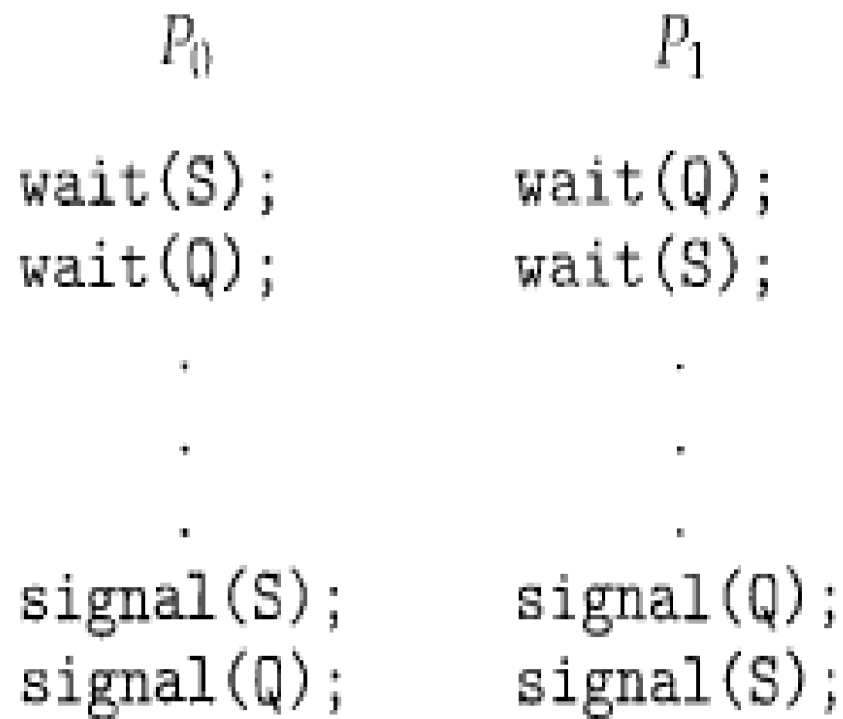
release lock

remainder section

} while (TRUE);

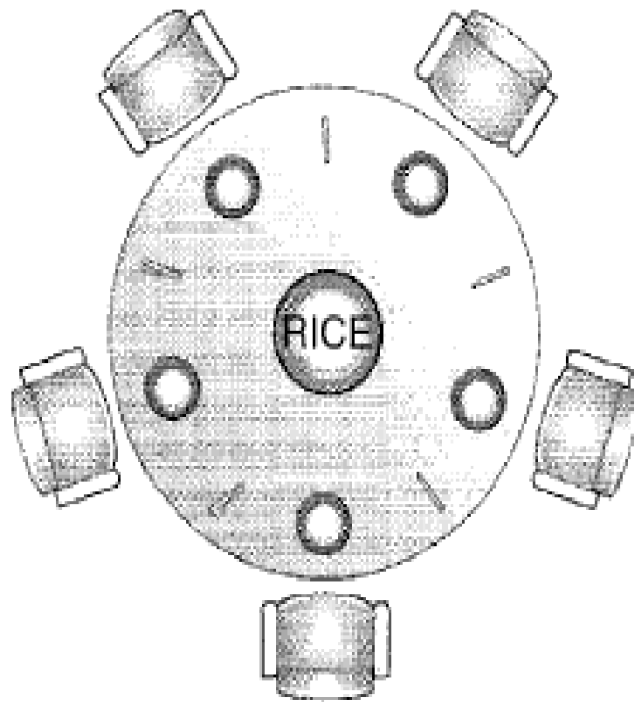
5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.



5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.

```

monitor dp
{
    enum {THINKING, HUNGRY, EATING}state[5];
    condition self[5];

    void pickup(int i) {
        state[i] = HUNGRY;
        test(i);
        if (state[i] != EATING)
            self[i].wait();
    }

    void putdown(int i) {
        state[i] = THINKING;
        test((i + 4) % 5);
        test((i + 1) % 5);
    }

    void test(int i) {
        if ((state[(i + 4) % 5] != EATING) &&
            (state[i] == HUNGRY) &&
            (state[(i + 1) % 5] != EATING)) {
            state[i] = EATING;
            self[i].signal();
        }
    }

    initialization_code() {
        for (int i = 0; i < 5; i++)
            state[i] = THINKING;
    }
}

```

5-Mark Explanation:

This diagram illustrates key operating system concepts from the course modules. It visually represents important architecture or mechanisms essential for understanding system operations.