

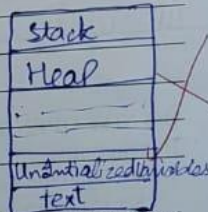
Q1.

[2 + 2 Points]

Give short answers in the space provided.

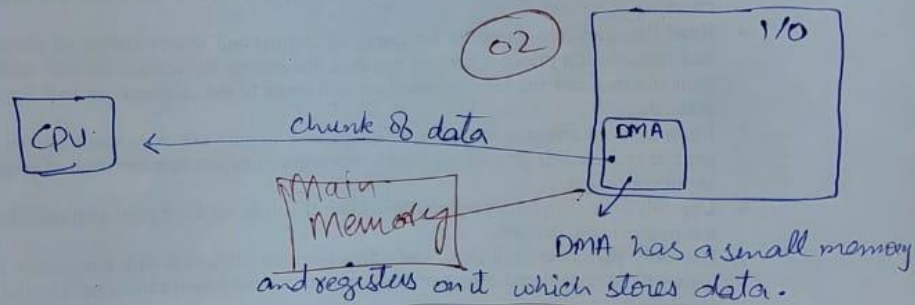
a. What is the process control block (PCB)? What is inside a PCB?

⑤ Each process in the system memory has a PCB. It is a block of memory which contains data and control instructions related to that process. Whenever a child process is created, a copy of this PCB is assigned to the child process.



b. How does DMA work in modern computers? Explain it with the help of a diagram. [3 + 3 Points]

Diagram



③ DMA stands for Direct Memory Access. Old computers had a problem with getting I/O from the I/O devices. They received data ~~bit~~ byte by byte which made the process pretty slow.

Modern computers came up with a solution to this problem called the DMA. Now every I/O device has its own DMA module with an embedded system built into it which gets the data and send it to the CPU in form of ^{chunks} chunks. So now instead of CPU getting input byte by byte, it receives it in the form of ~~small~~ chunks of data which makes the overall system faster.

c. What is context switching and why it is necessary? [2+2]

4 Whenever the CPU switches from one process to another, it saves its state to the memory and load another process into the registers. This process is called context switching. The time taken by this process is called context switching overhead. It is necessary as the CPU has to continue on the process later and to do so, it has to save its state somewhere.

d. Give two reasons why caches are useful. What problems do they solve? What problems do they cause? [2+2+2]

04 Caches are small memories built into the CPU. Caches are useful as they are relatively very fast memories than your system memory. So, they make accessing the running process data very fast.

If there were no cache, the system memory bandwidth would bottleneck the CPU speed. So, it solves this problem.

It causes context switching overhead as CPU has to load data into cache for every ~~process~~ running process.

e. What are the differences between a trap and an interrupt? What is the use of each function. [2+2]

There are two types of interrupts,

- 1) Hardware interrupt caused by external devices
- 2) Software interrupt caused by other processes.

1.5 Interrupt is used to stop the CPU from what it's doing and do something else. eg (receive I/O)

f. Why programmer prefer to write applications program according to API instead of system calls. Explain with reason. [4 pts]

4 API Provide a better and easier way to do stuff. Hence APIs are preferred by programmers rather than system calls.

System calls are mostly used for hardware level programming as they are faster.
4x3=12 points

Question 2

For each of the following questions four choices are provided, there is exactly one correct answer. You are required to encircle the correct choice.

12 (i) What does LINE X in the below program prints?

- a) 0, 1, 2, 3, 4
- b) 0, -1, -2, -3, -4
- ☒ c) 0, -1, -4, -9, -16
- d) 0, 0, 0, 0

0 -1 -4 -9 -16

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#define SIZE 5
int nums[SIZE] = {0, 1, 2, 3, 4};
int main()
{
    int i;
    pid_t pid;

    pid = fork();
    if (pid == 0) {
        for (i = 0; i < SIZE; i++) {
            nums[i] *= -1;
            printf("CHILD: %d ", nums[i]); /* LINE X */
        }
    }
    else if (pid > 0) {
        wait(NULL);
        for (i = 0; i < SIZE; i++)
            printf("PARENT: %d ", nums[i]); /* LINE Y */
    }
    return 0;
}
```

(ii) What does LINE Y in the above program prints?

- ☒ a) 0, 1, 2, 3, 4
- b) 0, -1, -2, -3, -4
- c) 0, -1, -4, -9, -16
- d) 0, 0, 0, 0

(iii) What does Line A in the below Program prints for Value1?

```
int value = 10, i = 0;
int main() {
    pid_t pid;
    pid = fork();
    if (pid == 0) {
        value += 15;
        printf("Value1 = %d", value); /*LINE A*/
    }
}
```

```
//Equivalent to cout << "Value1 = " << value ;
return 0;
}
else if (pid > 0) {
    wait(NULL);
    while(i < 3) {
        printf("Value2 = %d\n", value); /* LINE B */
        //Equivalent to cout << "Value2 = " << value << endl ;
        value = value + 3;
        i++ ;
    }
    return 0;
}
```

- a) 10
- b) 28
- ☒ c) 25
- d) 31

(iv) What does **Line B** in the above program prints for **Value2**?

- a) 13, 16, 19
- b) 25, 28, 31
- ☒ c) 10, 13, 16
- d) 28, 31, 34

Question 3 12 Points

1. Identify whether the following instructions are privileges or non-privileges instruction. [6 pts]

Instructions	Privileges/Non-Privileges
Generate any trap instruction	Privileged ✓
Clear the Memory or Remove a process from the Memory.	Privileged ✓
Set Timer	Non Privileged ✓
Sending final printout	Non Privileged ✓
I/O instructions	Privileged ✓
Halt instructions	Privileged ✓

2. Diagrammatically represents the process migrating among various queues. [6 pts]

