Name - Fardouse Lomat Jahan Rumpa

ID- 2219170041

Department - B.Sc in CSE (E)

Batch - 17th

Subject code - CSE - ~~334~~ 333

Subject title - Computer peripheals & Interfacing

Ans to the Qus NO: 01(a)

Answer: Computer pheripherals such as disk drives, printers etc. Work in different ways and linking a peripheral to the processor is a difficult task.

They work at different speeds, use different Codes transfer different amounts of data at a time, and even work at different voltages.

An Interface is the hardware and software needed between a processor and a peripheral device in order to compensate for differences in their operat characteristics. The interface allows the two devices to communicate correctly.

Example: Taking input from a keyboard is peripheral interfacing. Here keyboard is the periphera device. Printing a document with a printer is a peripheral interfacing where printer is the peripheral device. Showing a Video on the monitor is a periphera interfacing and this case monitor is the peripheral device.

Ans to the Qus NO: 01(b)

Answer: Interrupt: An interrupt is a signal sent to the processor that interrupts the current process. It may be generated by a hardware device or a software program.

A hardware interrupt is often

created by an input device such as a mouse or keyboard. For example, if you are using a word processor and press a key, the program must proceed the input immediately. Typing "hello" creates five interrupt requests, which allows the program to display the letters you typed. Similarly, each time you click a mouse button or tap on a touchscreen, you send an interrupt signal on the device.

Ans to the Qus NO: 02 (a)

Answer: When a Process is executed by the CPU and when a user Request for another Process then this will create disturbance for the Running Process. This is also called as the interrupt.

Types of interrupt:

Generally there are three types o interrupts those are Occured for example-

1) Internal interrupt.

2) Software Interrupt.

3) External interrupt.

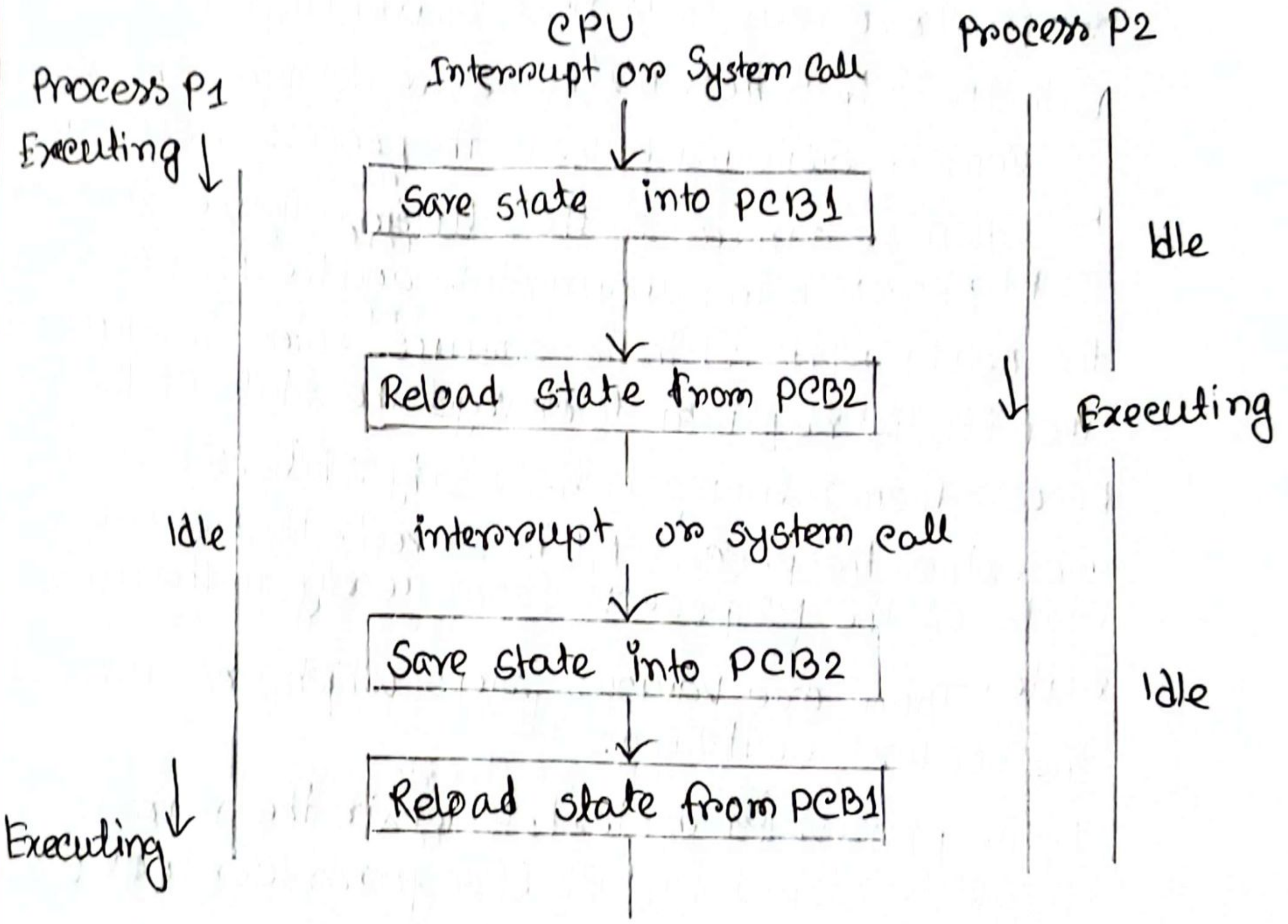
Ans to the Qus NO: 02(b)

Answer: Context switching machanism:

Context switching is the Switching of CPU from one process to anotherprocess. context switching means storing the process state so that we can reload the process when needed. and the execution of the process can be resumed from the same point later.
Context Switching is the charateristic of a multitasking operating system. In Context switching, One CPU can be shared among several processes. In other words, Context Switching is the mechanism that permits a Signle CPU to handle several threads or processes without the need for extra processors.
In context switching, processing are switched so quickly that the user gets the myth that all processes are running simultaneously.
But in the process of context switching, there are lots of steps that we need to follow. We cannot directly change or switch the process from running state to ready state. It is mandatory to save the context of that process. If we donot save the context of the process, while again executing the process, we need to start its execution from the beginning. In reality, the process from running state to ready state. It is mandatory to save the

②

context of that process. ~~If we do~~ we need to start its execution from the process in its preriouse execution. So, it is ~~ex~~required to save the context of the process before placing some other process in the running state. Context means data of CPU registers and program countere anytime.



Context switching Triggers:

The context switching Triggers are:

1. Intererrupts.
2. Multitasking.
3. kernal / User switch.

Interrupts: We require context switching if there is an interruption of CPU to get data from the disk read.

Multitasking: if the CPU has to move processes in and out of memory so that it can user mode to kernal mode.

Steps Involved in Context switching:

With the help of the below figure, we describe the procedure of context switching between the processes, which are P1 an P2. when we can see in the following figure that initially, the P1 process is in running state and the P2 process is in the ready state. If there occurs some interrupting. then its is required to change the state of the P1 process from running to the ready state. when the context of the process P1 is saved, then change the state of the P2 process from ready to the running state. There are various steps which are involved in the context switching.

1. The process P1 context, which is in the running state, we will be stored in PCB (Program Control Block). That is called PCB1.

2. Next, PCB1 is transferred to the appropriate queue, i.e, the I/O queue, ready queue, and the waiting queue.

3. Then from the ready queue, we choose the new process. which is to be executed. i.e, the process P2.

④ Next, we update the PCB (Program Control Block) of the P2 process called PCB2. It includes switching the process state from one to another (Ready, blocked, suspend or exit) if the CPU previously executed process P2, then we get the location of the last executed process so that we can again proceed with the P2 Process execution

⑤ In the same manner, if we again need to execute the process P1, then the same procedure is followed.

Ans to the Qus NO: 03 (a)

Answer: The basic concept of Analog interfacing:

There are may be several analog interfaces in the circuits such as ADC, DAC, Power management, grounds, etc. In Communication Circuits, there may be Oscillators, mixers, output amplifiers, buffers, speakers etc. Any terimial where an analog signal comes in or goes out and the hardware associated with that, can be consider as an analog interface.

Most of the time, the term is used to differentiate tha analog portion of the circuit with the digital portion.

Example:

An Interface on a circuit or a device such as a phone, which takes in an analog signal and then converts it to digital, can be refferred for the bes example for Analog interface.

Ans to the Qus NO: 03 (b)

Answer: The Interrupt Vector table (IVT) and Service Routine are key components of interrupt handing in computer Systems. They are responsible for managing and handling interrupts, which are asynchronous events that occur during the normal execution of a program and require immediate attention from the CPU.

The interrupt Vector Table (IVT) is a data structure that Contains a Collection of memory addresses, each associated with a specific interrupt type or event. These memory addresses point do the corresponding interrupt service routine (ISR) that should be executed when the associated interrupt occurs. The IVT is typically located in a reserved area of memory and is usually initialized during system boot-up.

The service Routine, also known as the interrupt Service Routine (ISR) or interrupt Handler, is a piece of code that is executed in response to a specific interrupt event. When an interrupt occurs, the CPU interrupts the normal execution of the currently running program and transfers control to the ISR pointed to by the corresponding entry in the IVT. The ISR then executes the necessary actions to handle the interrupt, such as

saving the state of the interrupted program, servicing the interecrupt request, and restoring the original program state before resuming its execution.
The works of the IVT and Service Routine are as follows

① IVT : The IVT is responsible for maintaining a table of memory addresses that point to the appropriate ISRs for different types of interrupts. It servies as a lookup table that allows the CPU to quickly locate the appropriate ISR to handle an interrupt event.
② Service Routine : The Service Routine (ISR) is Responsi-ble for actually handling the interrupt event. Its main tasks include saving the state of the interrupted program, Servicing the interrupt request, and restoring the original program state before return-ing control to the interrupted program, The specific work of the ISR depends on the type of interrupt being handled, such as handling I/O events, timer events or hardware faults.

In Summary, the IVT and Service Routine work together to manage and handle interrupts in a computer system. The IVT provides a lookup table of memory addresses that pointed to the appropriat ISRs for different interrupts, while the Service Routine is the code that actully executes the necessary actions to handle the interrupt event.

Ans to the QUS NO: 02 (e)

Ans: The term "control word" typically refers to a binary value or bit pattern that is used to control the behavior of a device or system. The interpretation and meaning of a control word would depend on the specific context in which it is used.

Based on the binary value "11010110" that you provided, here is a possible interepretation of it's control word based on the individual bits:

Bit 7 (leftmost)	1
Bit 6	1
Bit 5	0
Bit 4	1
Bit 3	0
Bit 2	1
Bit 1	1
Bit 0 (right most)	0

again, whitout knowing the specific context or system where this control word is used, it is not possible to provide a definitive interpretation. However, here is an example of a possible interpretation.

Ans to the Qus NO:03(C)

Answer: As an AI language model, I do not have real-time access to specific memory addssess as they are system-dependent and can change dynamically during runtime. Memory addressess are managed by the operating System and hardware of the computer or device in use. However, in general, CS 150 and IP 150 are not memory addressess, but rather course codes that could refer to computer systems, data and instruction are stored in memory using memory addresses, which are typically represented as hexadecimal numbers. The specific memory addresses where data or instructions are stored would depend on varioues factors such as the system architecture, operating System, and memory management techniques used. The exact memory addresses for storing CS 150 and IP 150, if they are used as data ar instructions. would depend on how they are utilizd within a specific computer System or g program