(1) Describe the procedure of an interrupt (i.e., describe the six steps in an interrupt).

① some event occurs at an I/O device (that requires a processor to take care of)

② an I/O device sends an interrupt signal to a processor

③ the processor performs a context switching (from a user program to the device driver of the I/O device)

④ the processor reads data from the I/O device (the processor does it using the device driver of the I/O device)

⑤ the processor saves data to the memory

⑥ the processor performs another context switching (from the device driver of the I/O device to the interrupted user program)

(2) What is “context switch”? What does a processor actually do during a context-switching?

- The context switching is the act of switching a processor from one to program to another (the act of stopping a processor from running the program it currently runs and running another program).

- In each context switching, a processor saves the processor registers of the program that is currently running to the stack area (and loads the copy of the processor registers of the program the processor starts running from the stack to the processor registers).
(3) What are the two primary advantages in using DMA for processing I/O events?

- Since the main processor no longer has to transfer data from an I/O device, handling data from I/O devices will not slow down execution of user programs.

- Since each data will be transferred through the system bus only one (compared to twice for interrupts), moving data (especially a large volume of data) from I/O devices to memory takes less time.

(4) What is “device driver”?

Device drivers are programs a processor needs for dealing with I/O devices (processors need a device driver for each I/O device). When processors are transferring data from an I/O device or when processors are sending data to I/O devices, processors need to do so by executing their device drivers.

(5) When an interrupt occurs while a user program is being executed by a processor, the stack area in the user program is used, but why?

For each interrupt, the stack area is used for a processor to save (keep) the processor registers of the currently running program. Processors need to do so before they start running another program since, when the processors need to resume those interrupted programs, the processors need the contents of their processor registers.