PES UNIVERSITY
(Established under Karnataka Act No.16 of 2013)
100-ft Ring Road, BSK III Stage, Bangalore – 560 085
Department of Computer Science & Engg

Session: Jan-May 2021

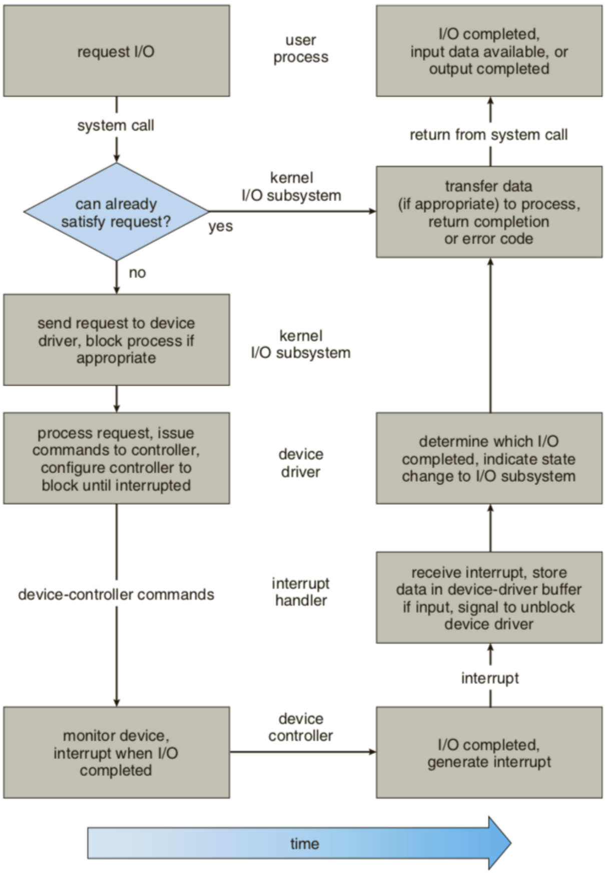
UE19CS254: Operating Systems
UNIT 5: Question Bank (Solutions to selected questions)

Chapter 13: I/O Systems

1 **Explain in detail the life cycle of an I/O request.**

Answer:

The typical life cycle of blocking read requests is shown in the following figure.



request I/O
user process
I/O completed, input data available, or output completed
system call
return from system call
can already satisfy request?
yes
kernel I/O subsystem
transfer data (if appropriate) to process, return completion or error code
no
send request to device driver, block process if appropriate
kernel I/O subsystem
process request, issue commands to controller, configure controller to block until interrupted
device driver
determine which I/O completed, indicate state change to I/O subsystem
device-controller commands
interrupt handler
receive interrupt, store data in device-driver buffer if input, signal to unblock device driver
interrupt
monitor device, interrupt when I/O completed
device controller
I/O completed, generate interrupt
time

System call - For every I/O request, process issues blocking read() system call to previously opened file descriptor of file. If data required is already available in buffer cache, it will be returned to process, and in that case I/O request is completed.

Alternative approach if input is not available –

If the data is not available in the buffer cache then physical I/O must be performed. The process is removed from the run queue and is placed on the wait queue for the device, and I/O request is scheduled. After scheduling, the I/O subsystem sends a request to the device driver via subroutine call or in-kernel message but it depends upon the operating system by which mode request will be sent.

Device Driver
After receiving the request, the device driver has to receive data. It will receive data by allocating kernel buffer space and after receiving data it will schedule I/O. After this, command will be given to the device controller by writing into device-control registers.

Device Controller
Now, device controllers operate device hardware. The data transfer is done by device hardware.

DMA Controller
After data transfer, the driver may poll for status and data, or it may have set up DMA transfer into kernel memory. The transfer is managed by a DMA controller. At last when transfer is complete, it will generate an interrupt.

Interrupt Handler
The interrupt is sent to the relevant interrupt handler through interrupt-vector table. It stores any necessary data, signals device drivers, and returns from interrupt.

Completion of I/O Request
Device driver receives a signal that the I/O request has completed. It also determines the request's status and signals to the kernel I/O subsystem that the request has been completed. After transferring data or return codes to address the space kernel, the process is moved from the wait queue back to the ready queue.

Completion of System Call
When the process moves to the ready queue, the process is unblocked. When the process is assigned to the CPU, the process resumes execution at completion of the system call.

2	**In most multi-programmed systems, user programs access memory through virtual addresses, while the operating system uses raw physical addresses to access memory. What are the implications of this design for the initiation of I/O operations by the user program and their execution by the operating system?** **Answer:** The user program typically specifies a buffer for data to be transmitted to or from a device. This buffer exists in user space and is specified by a virtual address. The kernel needs to issue the I/O operation and needs to copy data between the user buffer and its own kernel buffer before or after the I/O operation. In order to access the user buffer, the kernel needs to translate the virtual address provided by the user program to the corresponding physical address within the context of the user program's virtual address space. This translation is typically performed in software and therefore incurs overhead. Also, if the user buffer is not currently present in physical memory, the corresponding page(s) need to be obtained from the swap space. This operation might require careful handling and might delay the data copy operation.

Chapter 14: System Protection

3	**In a ring-protection system, level 0 has the greatest access to objects, and level n (where n > 0) has fewer access rights. The access rights of a program at a particular level in the ring structure are considered a set of capabilities. What is the relationship between the capabilities of a domain at level j and a domain at level i to an object (for j >i)?** **Answer:** Domain "j" is a smaller set inside of Domain "i".
4	**What protection problems may arise if a shared stack is used for parameter passing?** **Answer:** Add a counter and have it increase for that object after it is accessed. Before accessing an object check the counter.
5	**If all the access rights to an object are deleted, the object can no longer be accessed. At this point, the object should also be deleted, and the space it occupies should be returned to the system. Suggest an efficient implementation of this scheme.** **Answer:** Reference counts

6 **Why is it difficult to protect a system in which users are allowed to do their own I/O?**

Answer:

Any time we give control over to users we run the risk of them messing something up. As far as using hardware and processes for I/O operations there is a lot of management that needs to be done to make sure that each process and hardware piece is being used properly. If users have free use of I/O operations then we run the risk of tasks not being completed in the order that they should be, and also the integrity of the entire system being compromised.

In earlier chapters we identified a distinction between kernel and user mode where kernel mode is used for carrying out privileged operations such as I/O. One reason why I/O must be performed in kernel mode is that I/O requires accessing the hardware and proper access to the hardware is necessary for system integrity. If we allow users to perform their own I/O, we cannot guarantee system integrity

7 **Consider a computing environment where a process is given the privilege of accessing an object only n times. Suggest a scheme for implementing this policy.**

Answer:

Add an integer counter with the capability

Chapter 15: System Security

14 **What are the four levels where security measures must be taken?**
Answer:

- Security must occur at four levels to be effective:
- Physical
- Human
- Avoid social engineering, phishing, dumpster diving
- Operating System
- Network

8 **What is the most common technique for security attacks?**

Answer:

Masquerading - one participant in a communication pretends to be someone else (another host or person).

9	**Provide examples of at least three program threats.** **Answer:** 1. Trojan Horse 2. Trap Door 3. Logic Bomb 4. Stack and Buffer Overflow 5. Viruses
10	**Provide examples of at least two system and network threats.** **Answer:** 1. Worms 2. Port Scanning 3. DOS
11	**Make a list of six security concerns for a bank's computer system. Foreach item on your list, state whether this concern relates to physical,human, or operating-system security.** **Answer:** ● The system should be located in a safe location - Physical ● The location of the system should be well guarded - Human ● All operations should be recorded in a logsystem - Operating System, Human ● Backup the system frequency - Human ● The backup medias need to be protected - Human ● The operating system needs to be updated frequently for fixing the bugs - Operating system, Human. Other security concerns related to bank computer system are: 1. Data Breach:This is the most dangerous and most common security concern.Data can be leaked out by internal intruders or some external factors. This is basically related to physical security. 2. DDOS: Distributed Denial Of Service is the attack when computers are made unreachable on the internet by keeping them busy handling fake server requests. suppose a hacker can easily make the server busy handling fake requests.