Assignment 7

Network Programming(IS F462)

Name: Shreyas Bhat
ID: 2015A7PS0033G
Mail: f20150033@goa.bits-pilani.ac.in

Q1) There are 5 main types of system calls which can be used by a user program.

1) Process Control

There exist system calls such as **wait()**, **fork()**, **exec()** etc. The application is able to interact with the processes using **pids**. When **fork()** is called in C, we are invoking the glibc wrapper which actually call the system call **clone()**. System calls are generally implemented as software interrupts. Some of these system calls such as **wait()** are blocking in nature, whereas **fork()** returns immediately.

Advantages

- Helps create, exit and wait for processes. Helps when processes get stuck or need to be closed.

Disadvantages

- As system calls allow interaction with lower level primitives in the kernel space, misuse of these system calls can lead to problems.

2) File Manipulation

There exist system calls such as **read()**, **write()**, **close(), symlink()** etc. The application is able to interact with the files using **file descriptors**. For example, **read()** is used to read some file which is described by a file descriptor which is obtained by calling **open(). read(), write()** are in general slow system calls because the process needs to wait for the disk to read/write.

Advantages

- Programs generally need access to data that is stored in files, and these syscalls help read, write into these files.

Disadvantages

- As system calls allow interaction with lower level primitives in the kernel space, misuse of these system calls can lead to problems. For example, files might not be closed after opening. These system calls are generally slow.

3) Device Manipulation

A process may need several resources to execute - main memory, disk drives, access to files, and so on. If the resources are available, they can be granted, and control can be returned to the user process. Otherwise, the process will have to wait until sufficient resources are available. System calls like **ioctl()** , **read()**, **write()** help requisition devices. Sometimes, I/O devices are identified by special **file names, directory placement, or file attributes.** These syscalls are also slow in nature as they involve the usage of file descriptors. These syscalls might need some time for the disk/IO device to be available.

Advantages

- This is needed when processes need several resources such as Disk Drives, GPU , Main Memory etc.

Disadvantages

- Resources might not be closed correctly after being used by processes and might lead to open descriptors.

4) Information Maintenance

System calls help in making information maintenance such as get/set time or date, get/set data of system, processes, files or attributes of device. System calls such as **getpid()**, **alarm()**, **sleep()** which helps in the same. The OS keeps information about all its processes, and system calls are used to access this information. These syscalls such are not blocking in nature.

Advantages

- These system calls helps in keeping track of process information and helps communicate important data between the user and the OS.

Disadvantages

- These system calls have the potential to alter important bookkeeping information, so must be used carefully. If not, can lead to unexpected results.

5) Communication

Systems calls are used for communication as they help in creating and deleting communications, sending or receiving messages. They help in attaching or detaching remote devices and in transfer or status information. This is done mainly in two ways i.e. message passing and shared memory. System calls such as **pipe()**, **shmget()**, **nmap()** are used for the same. **shmget()** is a slow system call but non blocking, **pipe()** uses **read()** and **write()** calls which block depending upon whether the pipe is empty or not.

Advantages

- This is often needed when data needs to be conveyed/shared to other processes.

Disadvantages

- If pipes created between processes aren't closed they would be orphaned and would take up resources even if not being used by any process. Also, there are typical IPC problems that come up such as Dining Philosophers, Producer Consumer etc. Safeguards must be kept in place while performing IPC.

Q2) Yes this is possible as this does happen in the case of TCP and IP. IP being a best effort service and TCP adds reliability at the transport layer via various mechanisms that detect packet loss, congestion control etc.

Q3) If we use a ACK instead of a NACK we will need to add something like a sequence number which identifies the next packet that it is expecting. So, if a packet was corrupt the sequence number would not change and a ACK with the same sequence number would be sent to the sender.