



Network and Operating System(CS5001NI) Course Work 3

Submitted By

Student ID : Manish Giri

Student Name: 16034959

Submitted To: MukeshRegmi(Lecturer)

Submission Date: 04/22/2018

TABLE OF CONTENT FOR TASK A

INTRODUCTION	3
SOURCE CODE.....	4
TESTING VALID CASES.....	7
Test Case 1	7
Test case 2.....	8
Test Case 3	9
Test Case 4	10
Test Case 5	11
Testing of 5 invalid inputs.....	11
Test Case 1	11
Test Case 2	12
Test Case 3	12
Test Case 4	13
Test Case 5	14
Contains of player files	15
ConTENTs of player files	16
CONCLUSION	16

INTRODUCTION

Task A is predicated on UNIX operating system shell programming in Korn Shell. This task provides North American country the information on the UNIX operating system shell programming in UNIX system software package. Main goal of task A is to show and develop a UNIX operating system shell programme that implements interaction with the UNIX operating system setting in an exceedingly friendly manner and execute easy input/output operations. These simplicity and friendliness shall be achieved by mean of:

- Constructing an interactive request by using Unix commands.
- Checking user input for errors.
- Diagnosing these errors with clear messages.
- Testing the script
- Preparing report.

SOURCE CODE

```
Code=123
count=1
file="yes"
repeat="yes"
function scode
{
    echo "Enter Secret Code:" # Asking User to Enter Secret key
    read scode
    while [[ $scode != $Code && count -lt 3 ]]
    do
        echo "Enter Secret Code:"
        read scode
        count=$(( $count + 1 ))
    done

    if [ $count -gt 3 ]
    then
        exit
    fi
}
scode
Ask_name()          #Asking user to input his/her name and ID number and printing
execution date
{
    echo "Your ID is: $2"
    echo "Your name is: $1"
    echo "Date and Time of Execution=$(date)"
}
Ask_name $1 $2      # Calling Ask_name function
while [[ $repeat = "yes" ]]
do

    while [[ $file -eq "yes" ]]
    do
        Five_teams() # Asking user for Favorite circket team
        {
            echo "Choose anyone among these five circket team"
            echo "Type IND for India"
            echo "Type AUS for Australia"
            echo "Type ENG for England"
            echo "Type PAK for Pakistan"
            echo "Type BAN for Banglaesh"
            echo "Enter Country Code:"
            read ccode
            until [[ $ccode = "IND" || $ccode = "AUS" || $ccode = "ENG" || $ccode =
"PAK" || $ccode = "BAN" ]]
            do
                echo "Enter correct Country Code:" # user enters country code
```

```

        read ccode
        done
case $ccode in
IND)
    echo "India is the most loved circket team." #prints about India
    ;;
AUS)
    echo "Australia is the most world cup winning team." # prints about
Australia
    ;;
ENG)
    echo "England is the team where there are great players." # prints
about England
    ;;
PAK)
    echo "Pakistan is the team with legendary bowlers" # prints about
Pakistan
    ;;
BAN)
    echo "Bangladesh's batsman are one of the best batsman" # prints
about Bangladesh
    ;;
*)
    echo "Wrong Option please enter correct one" # prints if user
enter wrong code
    ;;
esac
}
Five_teams
circket_players()
{
    echo "choose your fav player" #Asking user to select fav player among
seven
    echo "Type Kh for Kholi"
    echo "Type Sm for Smith"
    echo "Type Ro for Root"
    echo "Type Wa for Wasim"
    echo "Type Ra for Rahman"
    echo "Type Dh for Dhoni"
    echo "Type Du for Duminy"
    echo "Choose any three players"
}
circket_players
read p1 p2 p3
Player="Select Player Code:"
select pcode in $p1 $p2 $p3
do
    if [[ $pcode = $p1 || $pcode = $p2 || $pcode = $p3 ]]
    then
        if [[ -r $pcode ]]

```

```

        then
            echo "Your Selected Player is: $pcode"
            cat $pcode
        file="yes"

    else
        echo "File does not Exist"
        file="no"

    fi

break

fi

done
if [[ $file = "no" ]]
then
    continue
else
    break
fi

done

echo "Do you want to repeat all Above Steps? yes/no:" # Asking user if he/she
wants to repeat all again
read repeat
done

```

TESTING VALID CASES

TEST CASE 1

Test Case No:1	
Purpose of Test Case: Ask user to input desired team code.	
Input Given:	IND
Output Expected:	Display information about Indian cricket team. Then ask user to input any three favorite player codes.
Actual Output:	Output Expected: Display information about Indian cricket team. Then ask user to input any three favorite player codes.
Observation: Output Expected: Display information about Indian cricket team. Then ask user to input any three favorite player codes. Therefore, this test is successfully done.	

Table 1: First testing table

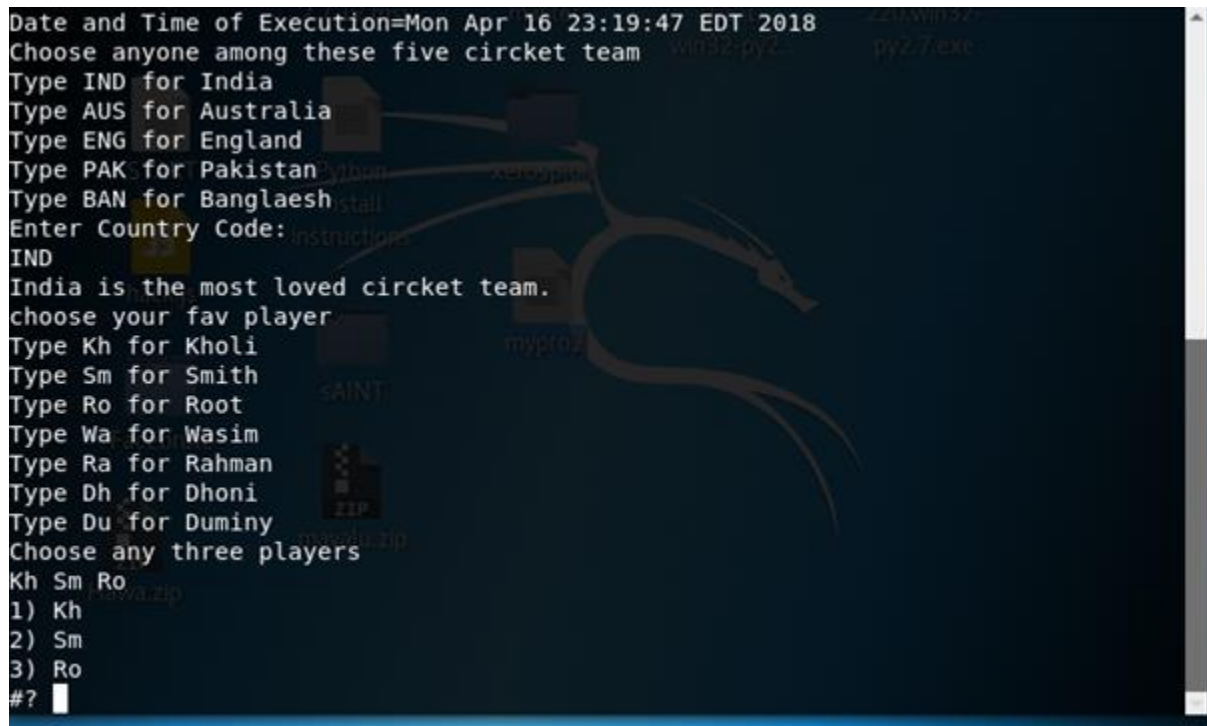


Figure 1: Screenshot of first testing case

TEST CASE 2

Test Case No:2	
Purpose of Test Case: Ask user to input any three favorite player codes.	
Input Given:	KhSm Ro
Output Expected:	The program to ask the user to select three players
Actual Output:	Ask user to select anyone player from selected three players.
Observation: Program has asked user to select anyone player from selected three players. Therefore, this test is successfully.	

Table 2: Second testing table



```

Date and Time of Execution=Mon Apr 16 23:19:47 EDT 2018
Choose anyone among these five circket team
Type IND for India
Type AUS for Australia
Type ENG for England
Type PAK for Pakistan
Type BAN for Banglaesh
Enter Country Code:
IND
India is the most loved circket team.
choose your fav player
Type Kh for Kholi
Type Sm for Smith
Type Ro for Root
Type Wa for Wasim
Type Ra for Rahman
Type Dh for Dhoni
Type Du for Duminy
Choose any three players
Kh Sm Ro
1) Kh
2) Sm
3) Ro
#?

```

Figure 2: Screenshot of second testing case

TEST CASE 3

Test Case No:3	
Purpose of Test Case: Display something about anyone selected player from chosen three players.	
Input Given:	Kh
Output Expected:	Description about Kholi
Actual Output:	Displays about the kholi.
Observation: Description of player viratkholi is displayed. Therefore this test is successful.	

Table 3: Third testing table

```

Type ENG for England
Type PAK for Pakistan
Type BAN for Banglaesh
Enter Country Code:
IND
India is the most loved circket team.
choose your fav player
Type Kh for Kholi
Type Sm for Smith
Type Ro for Root
Type Wa for Wasim
Type Ra for Rahman
Type Dh for Dhoni
Type Du for Duminy
Choose any three players
kh Sm Ro
1) kh
2) Sm
3) Ro
#? 1
Your Selected Player is: kh
echo"He is the best player of all time in Indian Cricet history."
Do you want to repeat all Above Steps? yes/no:

```

Figure 3:Screenshot of third testing case

TEST CASE 4

Test Case No:4	
Purpose of Test Case: Repeat steps from where the user is asked to input cricket team code.	
Input Given:	Yes
Output Expected:	To repeat the entire process again from where the user is asked to input any one cricket team code.
Actual Output:	Repeat steps from where the user is asked to input any one cricket team code.
Observation: Program asked users to input the code from the first. Therefore, this test is successful.	

Table 4: Fourth testing table

```

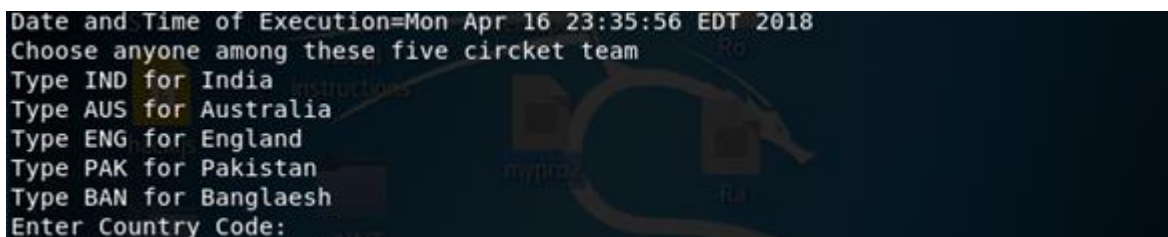
Type Sm for Smith
Type Ro for Root
Type Wa for Wasim
Type Ra for Rahman
Type Dh for Dhoni
Type Du for Duminy
Choose any three players
kh Sm Ro
1) kh
2) Sm
3) Ro
#? 1
Your Selected Player is: kh
echo "He is the best player of all time in Indian Cricet history."
Do you want to repeat all Above Steps? yes/no:
yes
Choose anyone among these five circket team
Type IND for India
Type AUS for Australia
Type ENG for England
Type PAK for Pakistan
Type BAN for Banglaesh
Enter Country Code:
  
```

Figure 4: Screenshot of fourth testing case

TEST CASE 5

Test Case No:5	
Purpose of Test Case: Ask user to input secret code to run the program.	
Input Given:	123
Output Expected:	To run the program and display the user's information.
Actual Output:	Runs the program, displays user ID, displays user name, displays date and time of execution and asks user to select anyone cricket team code from the list
Observation: As per the input given, the program has shown all the user's information. Therefore, the test is successful.	

Table 5: Fifth testing table



```
Date and Time of Execution=Mon Apr 16 23:35:56 EDT 2018
Choose anyone among these five circket team
Type IND for India
Type AUS for Australia
Type ENG for England
Type PAK for Pakistan
Type BAN for Banglaesh
Enter Country Code:
```

Figure 5: Screenshot of fifth testing case

TESTING OF 5 INVALID INPUTS

TEST CASE 1

Test Case No:1	
Purpose of Test Case: Show that program asks user to enter correct secret code when wrong code is entered.	
Input Given:	123
Output Expected:	To run the program and go further process.
Actual Output:	Asks user to input correct secret code again.
Observation: This given input 124 is in-correct so this program is asking again to enter the secret code.	

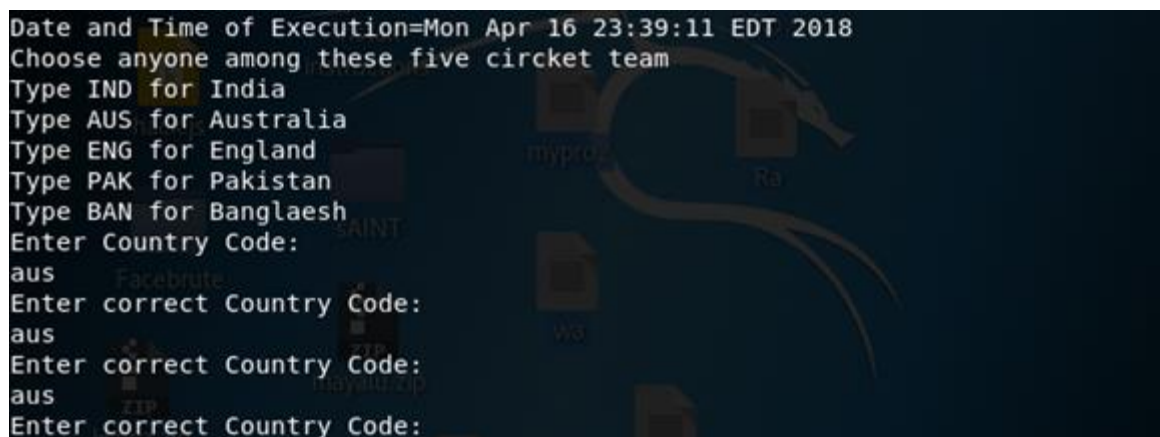
Table 6: Invalid testing case first



Figure 6: Screenshot of invalid testing case first

TEST CASE 2	
Test Case No:2	
Purpose of Test Case: Show that program asks user to enter correct country code when wrong code is entered.	
Input Given:	Aus
Output Expected:	To display about the country and then give option to select the favorite players.
Actual Output:	Asks user to input correct country code when wrong country code is entered.
Observation: The input in this case is Aus by which the expected result is not coming because for Australia the code AUS has been assigned.	

Table 7:Invalid testing case second



TEST CASE 3	
Test Case No:3	
Purpose of Test Case: Show that program displays error message when wrong player is selected.	
Input Given:	khsmro
Output Expected:	To display description of player kholi.
Actual Output:	Displays error message of File does not Exist when wrong player code is entered.

Observation: Displays error message because the file called kh has been deleted.

Table 8: Invalid testing case third

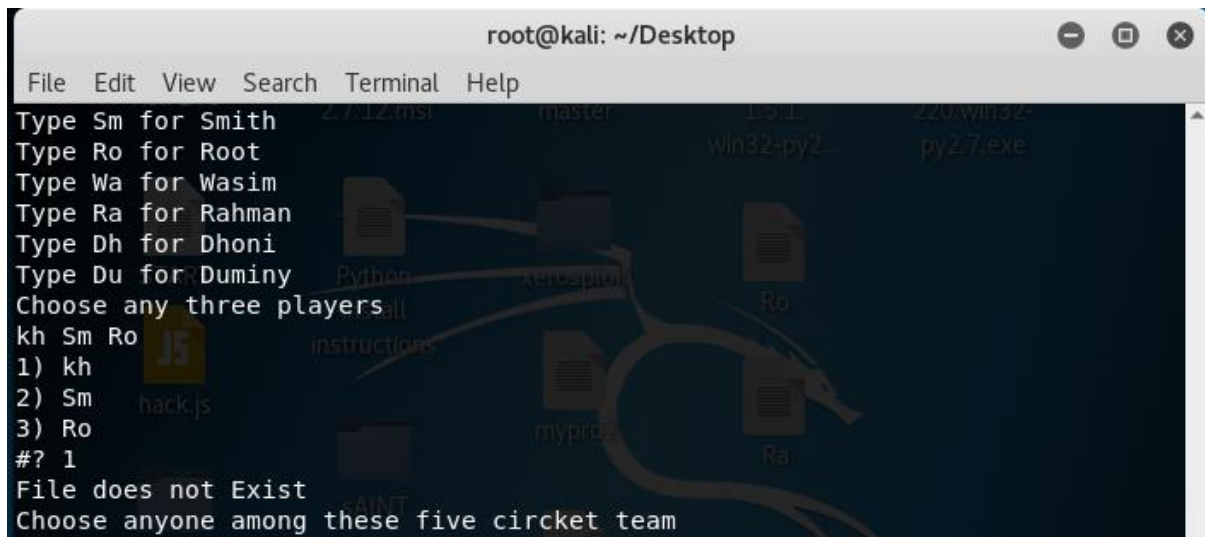


Figure 7: Screenshot of invalid testing case third

TEST CASE 4

Test Case No:4	
Purpose of Test Case: Show that program displays error message of File does not Exist when user selects player whose file is deleted and then repeats steps from where user is asked to enter any country code from the list.	
Input Given:	1 (Kh selected from three player KhSm Pa)
Output Expected:	To display the description of the player who has been assigned with Pa
Actual Output:	Displays error message of File does not Exist when user selects player and then repeats steps from where user is asked to enter any country code from the list.
Observation: This test is unsuccessful because a player with Pa has not been created.	

Table 9: Invalid testing case fourth

```

root@kali: ~/Desktop
File Edit View Search Terminal Help
India is the most loved circket team.
choose your fav player
Type Kh for Kholi
Type Sm for Smith
Type Ro for Root
Type Wa for Wasim
Type Ra for Rahman
Type Dh for Dhoni
Type Du for Duminy
Choose any three players
kh Sm Pa
1) kh
2) Sm
3) Pa
#? 3
File does not Exist

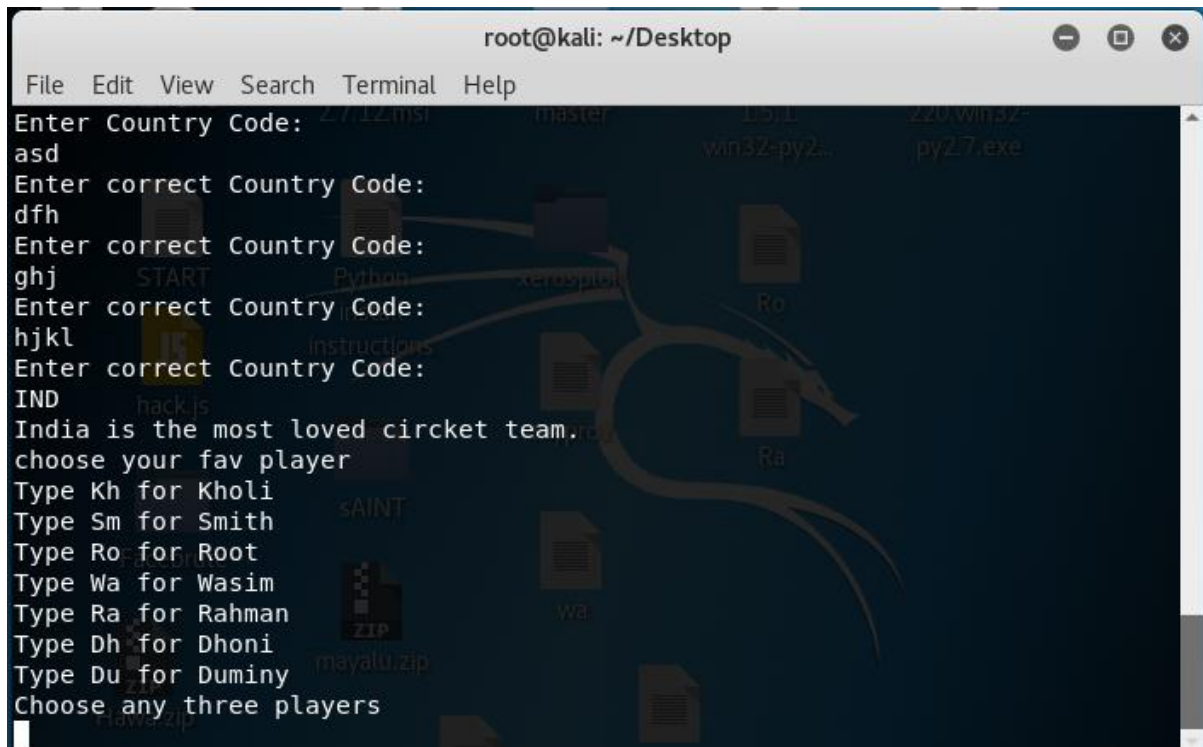
```

Figure 8: Screenshot of invalid testing case fourth

TEST CASE 5

Test Case No:5	
Purpose of Test Case: Show that program asks for country code until the user enters correct country code and program asks user to select any three players when correct code in entered.	
Input Given:	First – asd, second – dfh and third – ghj
Output Expected:	To display about the country as per the code entered.
Actual Output:	Asks user for correct country code until the user enters correct code then when correct code is entered program asks to select any three players.
Observation: Asks user for correct country code because a country with such code has not been assigned in the source code. Therefore, this test is un-successful.	

Table 10: Invalid testing case fifth



```
root@kali: ~/Desktop
File Edit View Search Terminal Help
Enter Country Code:
asd
Enter correct Country Code:
dfh
Enter correct Country Code:
ghj
Enter correct Country Code:
hjkl
Enter correct Country Code:
IND
India is the most loved circket team.
choose your fav player
Type Kh for Kholi
Type Sm for Smith
Type Ro for Root
Type Wa for Wasim
Type Ra for Rahman
Type Dh for Dhoni
Type Du for Duminy
Choose any three players
```

Figure 9: Screenshot of invalid testing case five

CONTAINS OF PLAYER FILES

Kholi (Kh):

Kholi is the best player of all time in Indian Cricket history.

Smith (Sm):

Smith is an Australian international cricketer and former captain of the Australian national Team

Root (Ro):

Joseph Edward Root is an English international cricketer, who is the current captain of the England test team.

Wasim (Wa):

Wasim Akram is a former Pakistani first-class cricketer, cricket commentator and television personality. He has made many records in his career.

Rahman (Ra):

Mustafizur Rahman is a Bangladeshi international cricketer.

Dhoni (Dh):

Dhoni is the Captain of National Team India. Dhoni is known as one of the most talented captain in cricket history till now. He is also known as game changer who took amazing decision in the field.

Duminy (Du):

Jean-Paul Duminy, often shortened to JP Duminy, is an international South African cricketer. He is vice-captain of the South Africa T20I team, and is currently standing in as captain, in the absence of the regular captain.

CONTENTS OF PLAYER FILES

Kholi (Kh):

Kholi is the best player of all time in Indian Cricket history.

Smith (Sm):

Smith is an Australian international cricketer and former captain of the Australian national Team

Root (Ro):

Joseph Edward Root is an English international cricketer, who is the current captain of the England test team.

Wasim (Wa):

Wasim Akram is a former Pakistani first-class cricketer, cricket commentator and television personality. He has made many records in his career.

Rahman (Ra):

Mustafizur Rahman is a Bangladeshi international cricketer.

Dhoni (Dh):

Dhoni is the Captain of National Team India. Dhoni is known as one of the most talented captain in cricket history till now. He is also known as game changer who took amazing decision in the field.

Duminy (Du):

Jean-Paul Duminy, often shortened to JP Duminy, is an international South African cricketer. He is vice-captain of the South Africa T20I team, and is currently standing in as captain, in the absence of the regular captain.

CONCLUSION

Task A is predicated on UNIX operating system shell programming in Korn Shell. This task provides North American country the information on the UNIX operating system shell programming in UNIX system software package. Main goal of task A is to show and develop a UNIX operating system shell programme that implements interaction with the UNIX operating system setting in an exceedingly friendly manner and execute easy input/output operations

TABLE OF CONTENTS FOR TASK B

Introduction.....	19
Aim:.....	19
Objective:.....	19
Network: Definition,Types,TopologIES	20
TOPOLOGIES	20
Network Connection: Protocols(TCP, UDP)	24
TCP	24
UDP.....	24
UNIX Communication:	25
Types of Unix:.....	26
Unix networking commands:.....	26
ifconfig	26
ping	26
traceroute	26
netstat	27
nslookup	27
whois	27
hostname	27
tcpdump	27
Security Aspects of Unix OS.....	27
1 - Common Sense Security.....	28
2 - File Permissions.....	28
Section 3 - Login daemons	29
Section 4 - Non-login daemons	29
I/O System Structure	30
I/O Traffic Controller	31
I/O Controller Functionality	32
I/O Scheduler	33

I/O Device Handler	34
Conclusion	36
REFERENCES.....	37

INTRODUCTION

Task B contents several topics. We have to research these topics and make report accordingly. It consists of UNIX OS, Networks and security issues of UNIX.

AIM:

The main aim of this coursework is to understand more about UNIX OS and its communication along with its security issues. It has also aimed to gain more knowledge about network protocols and topologies.

OBJECTIVE:

The main objective of this coursework is listed below.

- To know about security issues in UNIX
- To know more on UNIX communication
- To gain knowledge on protocols
- To know more different topologies
- To make better report

NETWORK: DEFINITION, TYPES, TOPOLOGIES

A network consists of multiple devices that communicate with each other. It may be as little as 2 computers or as massive as billions of devices. whereas a conventional network is comprised of desktop computers, trendy networks could embrace laptops, tablets, smartphones, televisions, play consoles, sensible appliances, and alternative physics.(Anon., n.d.)

Many types of networks exist, however they be 2 primary categories: LANs and WANs.

1. LAN

A local space network is proscribed to a particular space, like a home, office, or campus. A home network could have onerouter that gives each wired and wireless connections. as an example, a pc could connect with the router via local area network, whereas smartphones and tablets connect with the router via Wi-Fi. All devices connected to the router share constanetwork and sometimes constant web affiliation.(Anon., n.d.)

A larger network, like the network of an academic establishment, could also be comprised of the many switches, hubs, and local area network cables. it's going to conjointly embrace multiple wireless access points and wireless repeaters that offerwireless access to the network. whereas this kind of network is far additional complicated than a home network, it's still thought-about a LAN since it's restricted to a particular location.(Anon., n.d.)

2. WAN

A wide space network isn't restricted to one space, however spans multiple locations. WANs ar typically comprised of multiple LANs that ar connected over the web. a corporation WAN, as an example, could extend from the headquarters to differentoffices round the world. Access to WANs could also be restricted victimization authentication, firewalls, and different security measures. the web itself is that the largest WAN since it encompasses all locations connected to the web.(Anon., n.d.)

TOPOLOGIES

Topology may be physical or logical. topology is that the physical layout of nodes, workstations and cables within the network; whereas network topology is that the method info flows between completely different elements.(Anon., n.d.)

a. Bus Topology

Bus Topology is that the simplest of network topologies. during this form of topology, all the nodes (computers furthermore as servers) square measure connected to the only cable (called bus), by the assistance of interface connectors. This central cable is that the backbone of the network and is thought as Bus (thus the name). each digital computer communicates with the opposite device through this Bus.(Anon., n.d.)

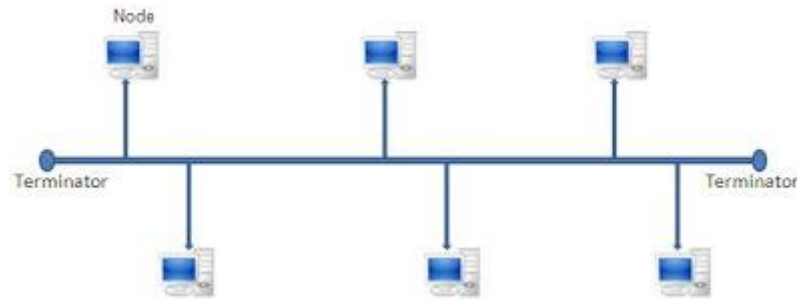


Figure 10:BusToploogy

b. Star Topology

In star, all the parts of network area unit connected to the central device known as “hub” which can be a hub, a router or a switch. in contrast to topology (discussed earlier), wherever nodes were connected to central cable, here all the workstations area unitconnected to central device with a point-to-point affiliation. therefore it may be aforementioned that each laptop is indirectly connected to each alternative node by the assistance of “hub”.(Anon., n.d.)

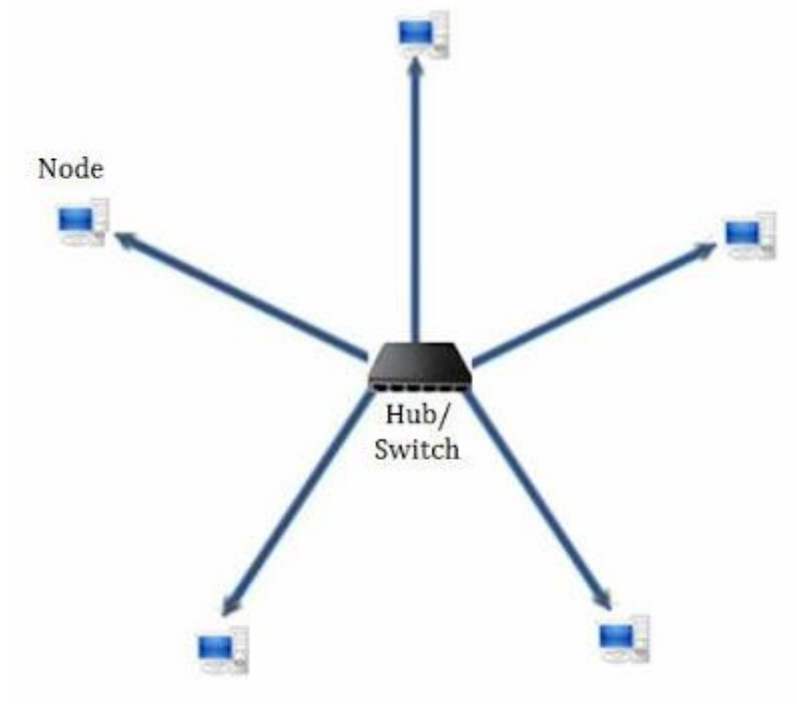


Figure 11:Star Topology

c. Ring Topology

In Ring Topology, all the nodes ar connected to each-other in such how that they create a control system. every digital computeris connected to 2 alternative elements on either facet, and it communicates with these 2 adjacent neighbors. information travels round the network, in one direction. causation and receiving of information takes place by the assistance of TOKEN.(Anon., n.d.)

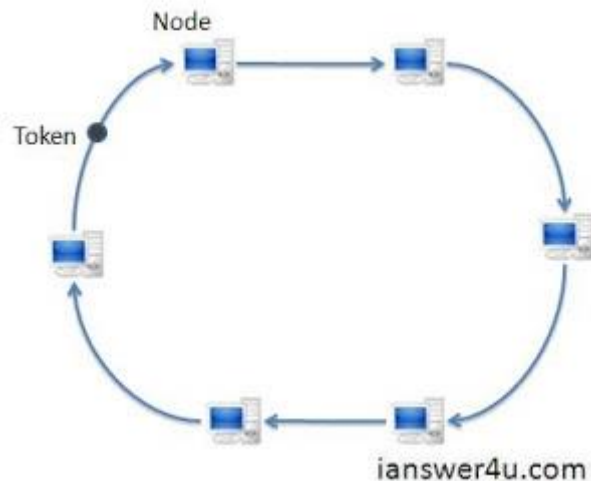


Figure 12: Ring Topology

d. Mesh Topology

In a mesh constellation, every of the network node, pc and alternative devices, ar interconnected with each other. each node not solely sends its own signals however conjointly relays information from alternative nodes. actually a real network topology is that the one wherever each node is connected to each alternative node within the network. this kind of topology is incredibly high-ticket as there ar several redundant connections, therefore it's not principally employed in pc networks. it's ordinarily employed in wireless networks. Flooding or routing technique is employed in network topology. (Anon., n.d.)

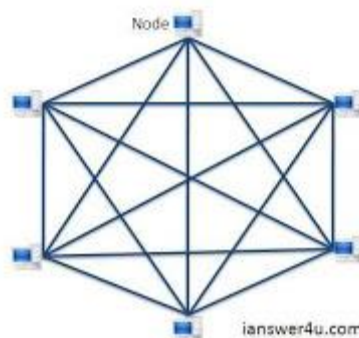


Figure 13: Mesh Topology

e. Tree Topology

Tree Topology integrates the characteristics of Star and bus. Earlier we tend to see however in Physical Star topology, computers (nodes) ar connected by one another through central hub. and that we conjointly saw in bus, work station devices ar connected by the common cable referred to as Bus. when understanding these 2 network configurations, we are able to perceive tree topology higher. In Tree Topology, the quantity of Star networks ar connected mistreatment Bus. This main cable feels like a main stem of a tree, and alternative star networks because

the branches. it's conjointly referred to as dilated star. LAN protocol is usually utilized in this kind of topology.(Anon., n.d.)

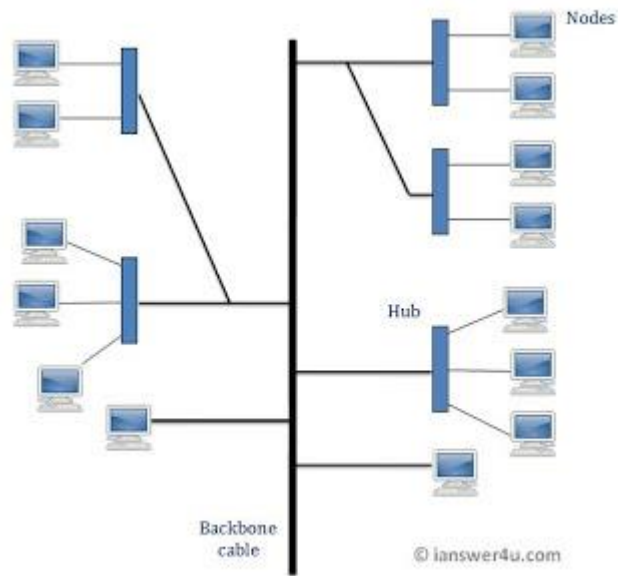


Figure 14:Tree Topology

« Back

NETWORK CONNECTION: PROTOCOLS(TCP, UDP)

TCP

The Transmission management Protocol (TCP) could be a connection-oriented reliable protocol. It provides a reliable transport service between pairs of processes capital punishment on finish Systems (ES) mistreatment the network layer service provided by the informatics protocol.(Anon., n.d.)

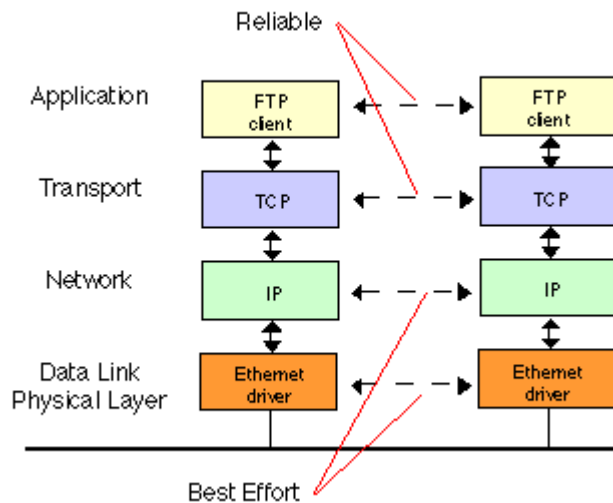


Figure 15:TCP providing reliable data transfer to FTP over an IP network using Ethernet

Transmission management Protocol is one in every of the foremost used protocols in digital network communications and is a component of the net protocol suite, usually referred to as the TCP/IP suite. Primarily, transmission control protocol ensures end-to-end delivery of knowledge between distinct nodes. transmission control protocol works unitedly with net Protocol, that defines the logical location of the remote node, whereas transmission control protocol transports and ensures that the information is delivered to the right destination.(Anon., n.d.)

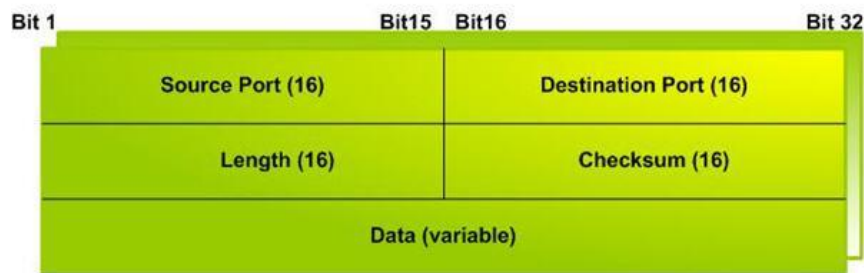
Before sending information, transmission control protocol creates a affiliation between the supply and destination node and keeps it live till the communication is active. transmission control protocol breaks massive information into smaller packets and additionally ensures that the information integrity is unbroken once it's reassembled at the destination node.(Anon., n.d.)

UDP

UDP stands for User Datagram Protocol — a datagram is the same thing as a packet of information. The UDP protocol works similarly to TCP, but it throws all the error-checking stuff out. All the back-and-forth communication and deliverability guarantees slow things down.

UDP is used when speed is desirable and error correction is not necessary. For example, UDP is frequently used for live broadcasts and online games.

The UDP Segment Format



TCP	UDP
Reliable	Unreliable
Connection-oriented	Connectionless
Segment retransmission and flow control through windowing	No windowing or retransmission
Segment sequencing	No sequencing
Acknowledge segments	No acknowledgement

UNIX COMMUNICATION:

UNIX is a software that was initially developed within the Sixties, and has been under constant development ever since. By software, we tend to mean the suite of programs that create the PC work. It's a stable, multi-user, multi-tasking system for servers, desktops and laptops.

UNIX systems even have a graphical interface (GUI) the same as Microsoft Windows that provides a simple to use surroundings. However, information of UNIX is needed for

operations which are not coated by a graphical program, or for once there's no windows interface out there, for instance, during a telnet session.

TYPES OF UNIX:

There are many alternative versions of OS, though they share common similarities. the foremost standard kinds of OS are Unix, Sun Solaris, GNU/Linux, and Mac OS .

Here within the faculty, we tend to use Solaris on our servers and workstations, and Stetson UNIX operating system on the servers and desktop PCs.

UNIX NETWORKING COMMANDS:

Here are some of the UNIX communicating commands. These commands help to communicate in UNIX.

IFCONFIG

One of the most basic networking commands is **ifconfig**. It will tell us about our network interfaces, the state that they're in, your assigned IP address(es), and even provide some counts of packets that have crossed the interface since the system was last booted. These days, we may see both ipv4 and ipv6 addresses.

PING

Ping is one of the most basic networking troubleshooting commands. It might tell us that we can reach a remote system, but we can't assume a system is down if we don't get a response -- especially if it belongs to some other company as the ping command and, often, all members of the icmp protocol family may be blocked somewhere along the route. If we get a response, we know the system we're pinging is up, but we also know that our network connection is good. I often ping a system somewhere else in the US (one on which I had an account many years ago) as a test of my network connection. If I can get a response from a system several states up the coast, I must be doing OK. If I'm not OK, then it's time to do a little troubleshooting.

The ping command generally behaves in one of two ways. It runs four times in succession and then tells you how many of the requests got through or it runs until you type ^c to stop it and then gives you the statistics. You can also provide the command with a -c option to specify how many times you want to ask. If I'm trying to verify that the systems that I'm expecting to find on a particular subnet are indeed up, I generally don't want to send each more than one ping.

TRACEROUTE

The traceroute command will attempt to provide a list of all the routers your connections cross when reaching out to a remote system. The output also provides some information on how long each segment of the path takes, thus giving you some notion of the quality of a connection.

NETSTAT

The `netstat` command can tell you about ongoing connections on the local system and ports (i.e., services) that are listening, indicating that services are waiting for requests to come through. By itself, `netstat` gives you a *lot* of information. With a `-a` option, it gives you even more.

NSLOOKUP

The `nslookup` command will do a DNS query for you. Assuming the system you are working on has a proper `/etc/resolv.conf` to let it know where to go to ask this type of question, it can help you find the IP addresses for particular systems and, often, the system names that go along with particular IP addresses. I've found that companies are often lax in setting up the reverse lookup (PTR) records that tie IP addresses back to hostnames, but they are wonderfully useful. By the time you are watching over hundreds or even thousands of servers, you're going to occasionally ask yourself something like "What system is 10.94.1.7?" and having an easy way to get an answer can be very helpful.

WHOIS

You can often find a lot more information about a domain using the `whois` command, the output below is truncated. Here I'm picking on my local gas station. Let's see what I can find out about Valero.

HOSTNAME

Generally, you know the name of the system you're logged into. After all, you just logged into it and, often, server names are set up as the system's command line prompt. But you might be logged into several systems at once or you may want to know just a bit more information. Here are some `hostname` command options that you might not be using.

TCPDUMP

The `tcpdump` command can print out the headers of network packets as they reach your server or can be used with various filters to select just the packets you want to see. You can also save packets for later analysis. I'll get into some interesting uses of `tcpdump` in my next post.

SECURITY ASPECTS OF UNIX OS

UNIX has been around for more than 10 years. Many operating systems have come along with new features in order to replace old operating system. But UNIX still remains dominant.

There are varieties of reasons for the popularity of UNIX. It is standardized, portable, and proven. Most importantly, however, UNIX is versatile. Most technological advances can be supported by the fundamental design of a UNIX system.

Unix is critical to business operations in most of the companies and Organizations Management tends to promote projects and people using the same technology the managers involved grew up with. I've looked and most express some security related concerns. In my experience most such clients have evolved IT infrastructures balancing large numbers of Windows or mainframe people against relatively few Unix people and consequently tilted much of their procedural decision making toward the policies appropriate to those environments and, correspondingly, not appropriate to Unix.

As a result, they very often institutionalize policies exemplifying one or more of what I think of as the four worst security strategies affecting Unix deployment in business and government.

1 - COMMON SENSE SECURITY

We have heard them so many times that we could recite them in your sleep. Rules like, "Don't write down your password." and, "Remember to log out." Do we need to repeat them? Yes. Security isn't about rules at all. Hackers don't sit down and follow a flowchart, so neither should we. Security is a way of thinking, a simple "What if?" that should accompany everything you do. Once you get used to it, it'll be easy.

Passwords: Programmers would like people to believe that if a potential intruder doesn't know a password there's nothing he can do. And banks would like you to believe that if a potential thief doesn't have your mother's maiden name, there's nothing he can do. Four nights ago I slept in a bunk above a man who robbed a bank with a broken, unloaded pellet gun. UNIX security, like any security, has many issues and scenarios to consider. So while the password isn't everything, it's a good place to start.

Whom to trust Admittedly, I'm bitter and cynical about the topic of trust. But again and again I have found systems that give user access-- or worse, superuser access-- to too many people. This creates many complications. When you give superuser access to anyone else, even someone who would never betray you, you are almost doubling your risk of an attack. Nothing is 100% secure, so it's simple probability. If your superuser account is 99% secure, and you make another, your level of security is 99% of 99% secure, or 98.01% secure.

Physical security When you think about computer security, you probably don't think about chains and padlocks. It is frequently overlooked, but I can't thoroughly address UNIX system security without mentioning it. If someone can physically access your computer, he or she can obtain superuser access. What's worse, he or she can probably even walk away with the computer. I'm not an expert on this aspect of computer security, but I have seen mistakes that you should avoid.

2 - FILE PERMISSIONS

Even small UNIX systems have thousands of files, and each file has associated with it a certain set of permissions.

the PATH: All major command shells have some notion of a path- a listing of directories to search for issued commands and includes /bin and /usr/bin. The problem here is that the path

is searched in order, with only the first match being executed. If a malicious user has write permission to a directory contained in the path, he or she can place trojans-- programs which look like something you want to run, but really aren't-- in the path for other users to stumble upon..

filesystems without permissions Some UNIX systems have the ability to mount a filesystem that does not use file permissions. Typically, a mount option specifies what permissions apply to all files on that filesystem. Sometimes, UNIX systems designed for multiple users run on computers that once were, or sometimes still are, used as personal computers. When a filesystem not made for multiple users is made accessible from UNIX, system administrators don't always stop to check whether or not any user can poke around there.

temporary files Many programs need temporary files for various purposes. Most of these are stored the /tmp directory, but temporary files can be anywhere the program has write access. Temporary files include not only short-term data storage, but also interprocess communication.

SECTION 3 - LOGIN DAEMONS

Login daemons are the primary ways to connect to your UNIX system. There are other ways, which we will examine later, but first we need to understand typical connections and how they can be abused. Login daemons typically wait for a connection, authenticate the connection, then provide some type of service.

TCP wrappers The biggest security advantage to an inetd approach is that it can support TCP wrappers. TCP wrappers are programs that act like traffic cops for incoming connections.

telnetd The most common way for logging in to a UNIX system is using telnet. Telnet simply establishes a TCP connection on port 23 and exchanges some data about terminal type. The login process is the same as a modem or text console.

ftpd The File Transfer Protocol (FTP) is TCP/IP's main method for transferring files between accounts. Unlike telnet and rlogin, ftp is considered a "non-interactive" login because programs are not executed. Restricting a user to only FTP logins, however, does not prevent a user from executing programs.

SECTION 4 - NON-LOGIN DAEMONS

httpd

. There are a lot of ways to use and abuse web pages, but here I'm only addressing ways in which a web server can be a risk for a UNIX system's security.

If you don't use inetd for your web server, and you want to use TCP port 80 (the standard http port), you'll need to run the server as root. (Ports 1 through 1023 are only usable by root.)

Fortunately, httpd provides a configuration file which lets you tell it what user to do its work as.

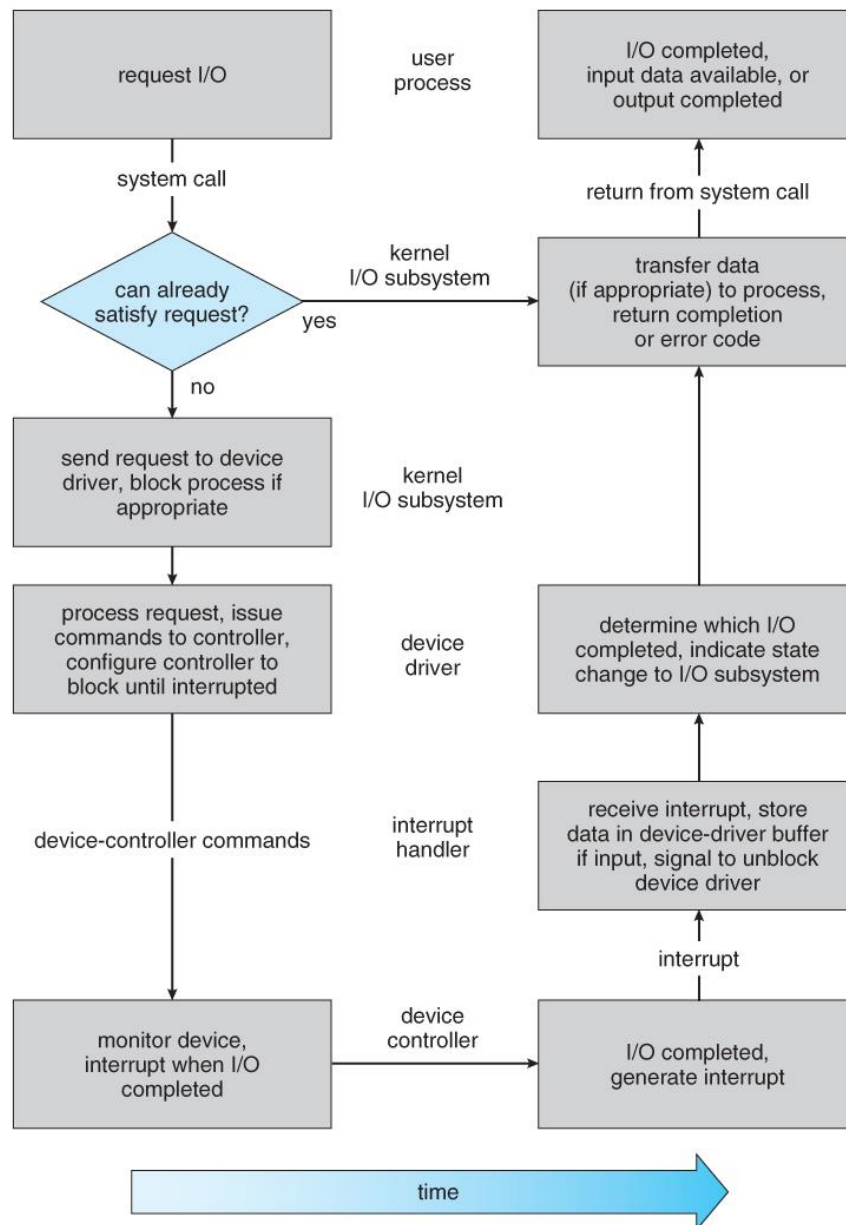
sendmail Equally as popular as httpd is sendmail. sendmail is the program that handles internet e-mail. There have been over a dozen major revisions in the sendmail program. Basically, when sendmail runs as a daemon, it listens on port 25 for incoming mail.

I/O SYSTEM STRUCTURE

Application programs can read from a device or write to a device with the help of operating system. For this process, I/O System structure is created to demonstrate the process.

Following are the components of an I/O Systems.

- I/O Hardware
- Application I/O Interface
- Kernel I/O Subsystem
- Transforming I/O Requests to Hardware Operations
- STREAMS
- Performance



I/O TRAFFIC CONTROLLER

Alternatively stated as input/output interface, IOC, or PIOC for Peripheral input/output controller. The input/output controller could be a device that interfaces between input or output device and therefore the laptop or hardware device. The input/output controller on a laptop is often placed on the motherboard. However, associate degree I/O controller also can be an inside add-on that may either be used as a replacement or to permit for extra input or output devices for the pc.

An operating system takes control by effectively managing userinput. It also manages both external and internal requests. This can include program and application requests, along

with information sent via keystrokes and other protocols. The main function of any operating system is to assist the CPU in facilitating user and program functions.

I/O CONTROLLER FUNCTIONALITY

• INTERFACE TRANSLATION

- connection, voltage
- protocol
- clocking

• ADDRESSING

The method used to pass signals from the CPU to the controller boards of peripheral devices on x86 machines. An I/O address, also called a "port address," references a separate memory space on PC peripheral boards, a little bit similar to memory-mapped peripherals that use blocks of memory. Peripherals often use both methods: an I/O address for passing control signals and memory for transferring data.

• MULTIPLEXING

When the TCP client is handling two inputs at the same time: standard input and a TCP socket, we encountered a problem when the client was blocked in a call to `fgets` (on standard input) and the server process was killed. The server TCP correctly sent a FIN to the client TCP, but since the client process was blocked reading from standard input, it never saw the EOF until it read from the socket (possibly much later).

We want to be notified if one or more I/O conditions are ready (i.e., input is ready to be read, or the descriptor is capable of taking more output). This capability is called **I/O multiplexing** and is provided by the `select` and `poll` functions, as well as a newer POSIX variation of the former,

• BUFFERING

Data from I/O devices often arrives at inconvenient times or in inconvenient units. The use of buffers in devices and in the kernel can hide some of this from the user. However, storing data in buffers can slow down I/O by requiring excessive copying, so there are trade-offs involved.

• ERROR DETECTION AND CORRECTION

- OS can recover from disk read, device unavailable, transient write failures.
- Most return an error number or code when I/O request fails.
- System error logs hold problem reports.

• CONTROL OF MULTIPLE STEPS

Usually, a controller interacts with the CPU in up to three ways:

- Through **control registers**. These act like memory locations, though they may be accessed through specialized I/O instructions (see below). For example, a keyboard device controller might have a write-only control register that sets the state of the LEDs and a read-only control register that holds the last character typed. A more sophisticated controller might have many control registers: a graphics controller might have many options for controlling scan timing, placement of data in memory, etc., that are all accessed through separate control registers.
- Through **shared memory**. The device controller might present some of its data as memory on the memory bus (e.g. the VGA display memory at 0xB8000). Or it might use **direct memory access** (see below).
- Through **interrupts**. Control registers and shared memory don't allow a device to signal when an operation has completed or data is ready. For this purpose, device controllers are given access to CPU interrupts (though usually via an intermediary **interrupt controller**).

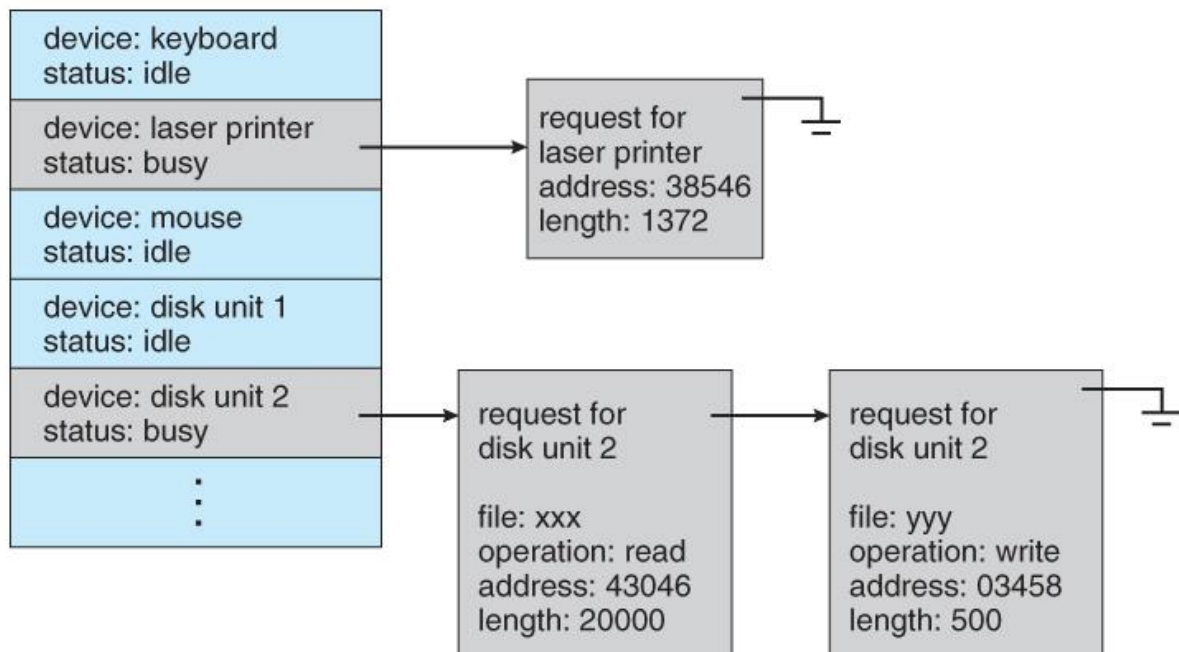
I/O SCHEDULER

Input/output (I/O) scheduling is that technique that PC operative systems use to make your mind up during which order the block I/O operations are going to be submitted to storage volumes. I/O programming is typically known as disk programming.

I/O scheduling sometimes must work with disk drives that have long access times for requests placed far-flung from this position of the disk head (this operation is termed a seek). To attenuate the impact this has on system performance, most I/O schedulers implement a variant of the elevator rule that reorders the incoming haphazardly ordered requests that the associated knowledge would be accessed with token arm/head movement.

I/O schedulers will have several functions counting on the goals; common functions embody the following:

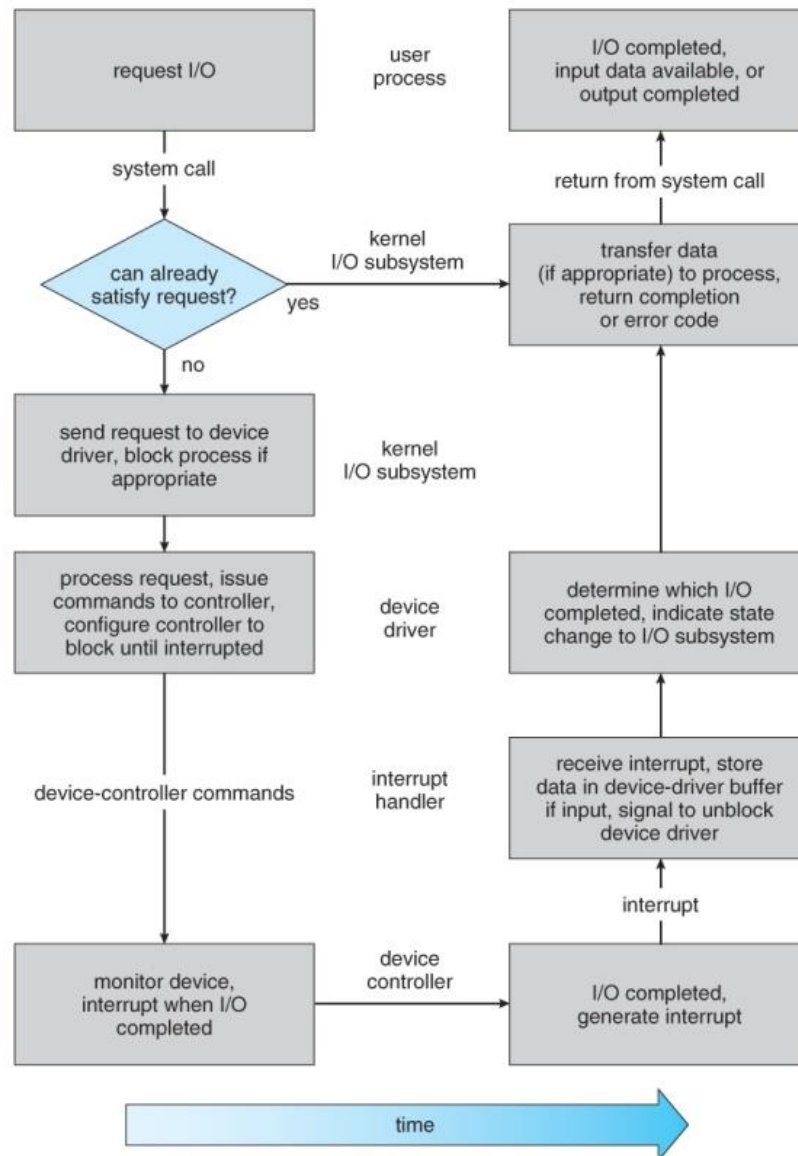
- to attenuate time wasted by disk seeks
 - To rank an exact processes' I/O requests
 - to grant a share of the disk information measure to every running method
 - to ensure that bound requests are going to be issued before a specific point in time
- Scheduling I/O requests can greatly improve overall efficiency. Priorities can also play a part in request scheduling.
 - The classic example is the scheduling of disk accesses
 - Buffering and caching can also help, and can allow for more flexible scheduling options.
 - On systems with many devices, separate request queues are often kept for each device:



I/O DEVICE HANDLER

The I/O library subroutines can send data to or from either devices or files. The system treats devices as if they were I/O files. For example, you must also open and close a device just as you do a file.

Some of the subroutines use standard input and standard output as their I/O channels. For most of the subroutines, however, you can specify a different file for the source or destination of the data transfer. For some subroutines, you can use a file pointer to a structure that contains the name of the file; for others, you can use a file descriptor (that is, the positive integer assigned to the file when it is opened).



CONCLUSION

Lastly, I even have conjointly explained the communication terms and commands of UNIX system OS along side elaborated expiation. I even have conjointly delineate concerning completely different protocols and topologies. I even have given short descriptions on I/O control, I/O device handler so on. These ar the items I even have complete in my report of TASK B. This report has given the step wise data concerning UNIX system OS, UNIX system communication along side its security problems and countermeasures.

REFERENCES

Anon., 2016. *Tech-Target*. [Online] Available at: <https://searchnetworking.techtarget.com/definition/Ethernet>.

computernetworkingsimplified, 1994. *computernetworkingsimplified.in*. [Online] Available at: <http://computernetworkingsimplified.in/data-link-layer/components-data-link-layer-llc-mac/>.

omnisecu, 1994. *www.omnisecu.com*. [Online] Available at: <http://www.omnisecu.com/tcpip/tcpip-model.php>.

studytonight, 1992. <https://www.studytonight.com/>. [Online] Available at: <https://www.studytonight.com/computer-networks/osi-model-network-layer>.

webopedia, 1997. *www.webopedia.com*. [Online] Available at: https://www.webopedia.com/quick_ref/OSI_Layers.asp.

(n.d.). Retrieved from <https://www.ianswer4u.com/2011/05/bus-topology-advantages-and.html>

(n.d.). Retrieved from <https://www.ianswer4u.com/2011/05/star-topology-advantages-and.html>

(n.d.). Retrieved from <https://www.ianswer4u.com/2011/05/ring-topology-advantages-and.html>

(n.d.). Retrieved from <https://www.ianswer4u.com/2011/05/mesh-topology-advantages-and.html>

(n.d.). Retrieved from <https://www.ianswer4u.com/2012/01/tree-topology-advantages-and.html>

(n.d.). Retrieved from <http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/tcp.html>

(n.d.). Retrieved from <https://www.techopedia.com/definition/5773/transmission-control-protocol-tcp>

TCP	UDP
Reliable	Unreliable
Connection-oriented	Connectionless
Segment retransmission and flow control through windowing	No windowing or retransmission
Segment sequencing	No sequencing
Acknowledge segments	No acknowledgement

