USN										

Internal Assessment Test 2 –DEC 2022
Scheme of Evaluation

Sub:	UNIX Programming					Sub Code:	18CS56	Branch:	ISE
Date:	02/12/2022	Duration:	90 min	Max Marks:	50	Sem/Sec:	V / A, B & C		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT

1 (a)	**Explain how test statement is applicable to numeric and string comparisons.**
	Scheme: test statement with numeric and string comparisons with examples-3+2 Marks
	Solution:
	✓Test statement is used to handle the true or false value returned by expressions, and it is not possible with if statement.
	✓Test works in three ways:
	1. Compare two numbers
	2. Compares two strings or a single one for a null value
	3. Checks files attributes
	Ex:
	`$ x=5;y=7;z=7.2`
	`1. $test $x -eq $y; echo $?  1  Not equal`
	`2. $test $x -lt $y; echo $?  0  True`
	`3. $test $z -gt $y; echo $?  1  7.2 is not greater than 7`
	Test — True if
	s1=s2 — String s1=s2
	s1!=s2 — String s1 is not equal to s2
	-n stg — String stg is not a null string
		[05]	CO2	L2

(b)	**Explain set and shift command in Unix to manipulate positional parameters.**
	Scheme: Explanation of set & shift command with examples- 4+6 Marks
	Solution:
	✓The set statement assigns positional parameters $1, $2 and so on, to its arguments. This is used for picking up individual fields from the output of a program.
	`ise@DESKTOP-ABO41VE ~`
	`$ set 9876 2345 6213`
	`ise@DESKTOP-ABO41VE ~`
	`$ echo "\$1 is $1,\$2 is $2,\$3 is $3"`
	`$1 is 9876,$2 is 2345,$3 is 6213`
	`ise@DESKTOP-ABO41VE ~`
	`$ echo "The $# arguments are $*"`
	`The 3 arguments are 9876 2345 6213`
	✓Shift transfers the contents of positional parameters to its immediate lower numbered one. This is done as many times as the statement is called. When called once, $2 becomes $1, $3 becomes S2 and so on.
		[05]	CO2	L2

2	**Explain any 5 general File APIs with an example.** **Scheme:** Explanation of any 5 File APIs with examples- 2+2+2+2+2 Marks **Solution:**	[10]	CO3	L2

```
#include < sys/types.h>
#include <unistd.h>
#include <fcntl.h>

int open(const char *path_name, int access_mode, mode_t permission);
```

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

int creat(const char *pathname, mode_t mode)
```

Equivalent to:

```
open(pathname, O_WRONLY|O_CREAT|O_TRUNC, mode)
```

```
#include <unistd.h>
#include<sys/types.h>

ssize_t read(int fd, void *buff, size_t size);
```

```
#include <sys/types.h>
#include <unistd.h>

ssize_t write (int fdesc , const void* buf, size_t size);
```

```
#include <unistd.h>

int close (int fdesc);
```

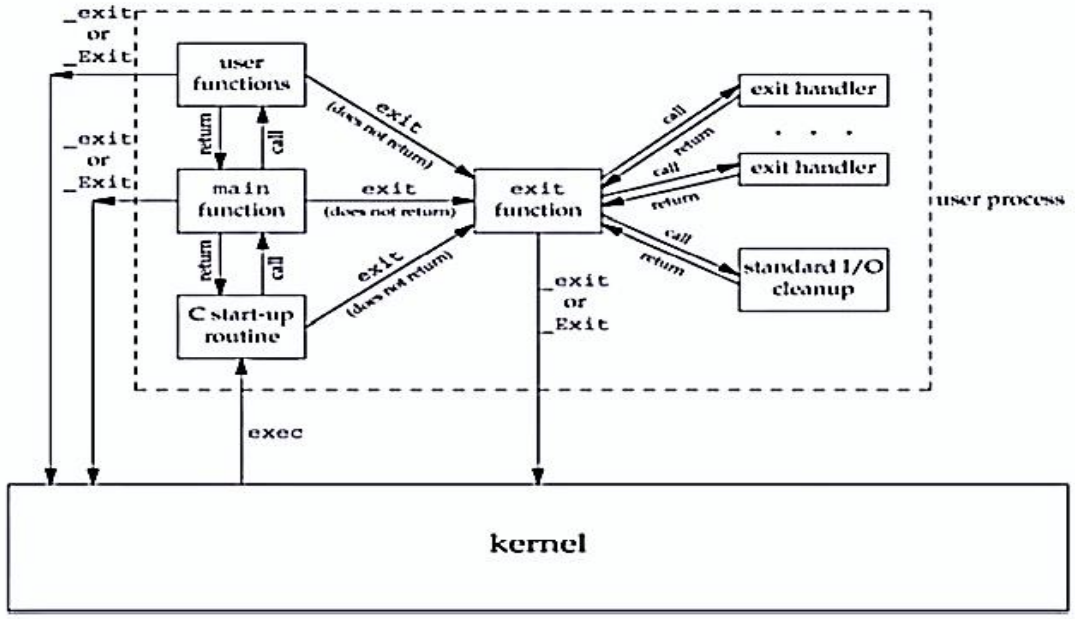
```
#include<unistd.h>
#include<sys/types.h>
#include<stdio.h>

int main()
{
    int n, fd;
    char buff[50];
    printf("Enter text to write in the file:\n");
    n= read(0, buff, 50);

    fd=open("file",O_CREAT | O_RDWR, 0777);

    write(fd, buff, n);
    write(1, buff, n);

    close(fd);
    return 0;
}
```

3	**With an example, explain grep command with all options.** **Scheme:** Explanation of grep command + any 6options with examples- 4+6 Marks **Solution:** ✓grep scans its input for a pattern displays lines containing the pattern, the line numbers or filenames where the pattern occurs. $grep options pattern filename(s)	[10]	CO2	L2

Option	Significance
-i	Ignores case for matching
-v	Doesn't display lines matching expression
-n	Displays line numbers along with lines
-c	Displays count of number of occurrences
-l	Displays list of filenames only
-e exp	Matches multiple patterns
-f filename	Takes patterns from file, one per line
-E	Treats patterns as an ERE
-F	Matches multiple fixed strings

```
ise@DESKTOP-ABO41VE ~
$ grep "sales" emp.lst
2233|a.k.shukla|g.m.|sales|12/12/52|6000
1006|chanchal singhvi|director|sales|03/09/38|6700
1265|s.n.dasgupta|manager|sales|12/09/63|5600
2476|anil aggarwal|manager|sales|01/05/59|5000

ise@DESKTOP-ABO41VE ~
$ grep president emp.lst

ise@DESKTOP-ABO41VE ~
$ grep 'jai sharma' emp.lst
9876|jai sharma|director|production|12/03/50|7000

ise@DESKTOP-ABO41VE ~
$ grep "jai sharma" emp.lst
9876|jai sharma|director|production|12/03/50|7000

ise@DESKTOP-ABO41VE ~
$ grep -i 'agarwal' emp.lst
3564|sudhir Agarwal|executive|personnel|06/07/47|7500

ise@DESKTOP-ABO41VE ~
$ grep -n 'marketing' emp.lst
3:5678|sumit chakrobarty|d.g.m.|marketing|19/04/43|6000
11:6521|lalit chowdury|director|marketing|26/09/45|8200
14:2345|j.b.saxena|g.m.|marketing|12/03/45|8000
15:0110|v.k.agrawal|g.m.|marketing|31/12/40|9000

ise@DESKTOP-ABO41VE ~
$ grep -c 'director' emp.lst
4

ise@DESKTOP-ABO41VE ~
$ grep -c 'director' emp*.lst
emp.lst:4
emp2.lst:3

ise@DESKTOP-ABO41VE ~
$ grep -l 'marketing' *.lst
emp.lst
emp2.lst

ise@DESKTOP-ABO41VE ~
$ grep -e "Agarwal" -e "aggarwal" -e "agrawal" emp.lst
2476|anil aggarwal|manager|sales|01/05/59|5000
3564|sudhir Agarwal|executive|personnel|06/07/47|7500
0110|v.k.agrawal|g.m.|marketing|31/12/40|9000
```

4 (a)	**Write a menu driven shell script which outputs the following:** **i. List of files ii. Current system display settings iii. Present working directory iv. HOME directory of a user v. Terminal characteristics vi. process status** **Scheme:** shell script to output the given details – 6 Marks **Solution:** #!/bin/sh echo "MENU\n 1. List of files\n 2. Current System display settings\n 3. Present working directory\n 4. HOME directory of a user \n 5. Terminal characteristics\n 6. process status\n Enter your option:\c" read choice case "$choice" in ls –l ;; stty ;; pwd ;; echo "$HOME" ;; tty ;; ps ;; echo "invalid option" esac	[06]	CO2	L3
(b)	**Write a shell script to display the processes in the system every 30 seconds for five times using while loop.** **Scheme:** shell script to output the given details – 4 Marks **Solution:** #!/bin/sh echo " processes in system" cnt=5 while [$cnt –ne 0] do ps echo "===========" sleep 30 cnt=`expr $cnt -1` done	[04]	CO2	L3
5	**Illustrate the mechanism of how a C Program is started and terminated.** **Scheme:** Explanation of main function+Process Termination+ exit function+ Diagram– 2+4+2+2 Marks **Solution:** int main(int argc, char *argv[]); ✓There are eight ways for a process to terminate. ❑Normal termination occurs in five ways: 1. Return from main 2. Calling exit 3. Calling _exit or _Exit 4. Return of the last thread from its start routine 5. Calling pthread_exit from the last thread	[10]	CO3	L2

	❑Abnormal termination occurs in three ways:
	1. Receipt of a signal
	2. Response of the last thread to a cancellation request
	3. Calling abort function
	#include <stdlib.h>
	void exit(int status);
	void _Exit(int status); void _exit(int status);
	 			

6 (a)	**Write a short note on HERE (<<) document and logical operators.** **Scheme:** Explanation of Logical && and || – 4 Marks **Solution:** o Shell uses << symbols to read data from the same file containing the script – here document o Signifies: Data is here rather than a separate file. **Syntax:** command << LABEL • reads following lines until line starting with "LABEL" **Example:** % wc -l << DONE > line one > line two > DONE 2	[05]	CO2	L1

(b)	**With an example, explain Logical operators in shell programming.**
	Scheme: Explanation of Logical && and || – 4 Marks
	Solution:
	✓The shell provides two operators that alows conditional execution, the && and ||.
	Usage:
	cmd1 && cmd2
	cmd1 || cmd2
	✓&& delimits two commands. cmd 2 executed only when cmd1 succeeds.
	The logical OR , cmd2 gets executed only when cmd1 fails.
	`ise@DESKTOP-ABO41VE ~`
	`$ grep 'director' emp.lst && echo "pattern found"`
	`9876|jai sharma|director|production|12/03/50|7000`
	`2365|barun sengupta|director|personnel|11/05/47|7800`
	`1006|chanchal singhvi|director|sales|03/09/38|6700`
	`6521|lalit chowdury|director|marketing|26/09/45|8200`
	`pattern found`
	`ise@DESKTOP-ABO41VE ~`
	`$ grep 'manager' emp2.lst || echo "pattern not found"`
	`pattern not found`
	`ise@DESKTOP-ABO41VE ~`
	`$ grep 'marketing' emp2.lst || echo "pattern not found"`
	`6521|lalit chowdury|director|marketing|26/09/45|8200`
	`2345|j.b.saxena|g.m.|marketing|12/03/45|8000`
		[05]	CO2	L2

Faculty Signature CCI Signature HOD Signature