Output:

```
#gcc scall.c
#vi mfile.txt
#gcc scall.c
#./a.out
File Opened successfully
NCMS is written to file.
```

Expt. No. 11 Date 08/11/24

Page No. 20

PART - B

11) Write a program using system call: creat, open, write, close, stat, fstat, lseek.

vi scall.c

Program:

```
#include<stdio.h>
#include<fcntl.h>
int main()
{
int fd;
char buffer[100];
static char message[] = "NCMS";
fd = open("mfile.txt", O_RDWR);
if (fd != -1)
{
printf("File opened successfully\n");
write (fd, message, sizeof(message));
lseek (fd, 0, 0);
read (fd, buffer, size of (message));
printf("%s is written to file\n", message);
close(fd);
}
return 0;
}
```

Teacher's Signature

Output:

```
# gcc child.c
# ./a.out
Hello World
I am a child process
I am parent process.
```

Date 08/11/24

Expt. No. 12

Page No. 21

12) Write a program to create a child process and allow the parent to display "parent" and the child to display "child" on the screen

vi child.c

Program:

```
#include<stdio.h>
#include<sys/wait.h>
#include<stdlib.h>
int main()
{
int pid, status;
printf("Hello World\n");
pid = fork();
if (pid==-1)
{
perror("Bad fork\n");
exit(1);
}
if (pid==0)
printf("I am a child process\n");
else
{
wait(&status);
printf("I am parent process\n");
return 0;
}
}
```

Teacher's Signature

Output:

```
#gcc zombie.c
#./a.out
I am the child process with pid = 10232
The parent id is 10231
I am the parent process with pid = 10231
child pid is 10232
```

Expt. No. 13 Date 08/11/24

Page No. 22

13) Write a program to create Zombie process.

vi zombie.c

Program:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main()
{
pid_t P;
P=fork();
if(P==0)
{
printf("I am the child process with pid=%d\n", getpid());
printf("The parent id is %d\n", getppid());
}
else
{
sleep(3);
printf("I am the parent process with pid=%d\n", getpid());
printf("child pid is %d\n", P);
}
return 0;
}
```

Teacher's Signature

Output:

```
# gcc pipe.c
# ./a.out
Pass the values to child
child process received
hello
```

Date 08/11/24

Expt. No. 14

Page No 23

14) Write a program to implement inter-process communication using pipes

vi pipe.c

Program:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main()
{
pid_t p;
int fd[2], n;
char buffer[100];
pipe(fd);
p = fork();
if(p==0)
{
printf("Pass the values to child\n");
write(fd[1], "hello\n", 6);
}
else
{
sleep(3);
printf("child process received\n");
n = read(fd[0], buffer, 100);
write(1, buffer, n);
}
return 0;
}
```

Teacher's Signature

```
# cat > vehicle2
Jeep Car
# cat > vehicle3
```

Output:

```
# gcc copy.c
# ./a.out
  Bad address
# ./a.out vehicle2 vehicle3
  Jeep Car.
```

Date 09/11/24

Expt. No. 15 Page No. 24

15 Write a program to copy a file into another file using system calls.

vi copy.c

Program:

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
void typefile(char *fname)
{
int fd1, fd2;
char buffer[100];
fd1 = open(fname, O_RDONLY);
if(fd1 == -1)
{
perror(fname);
return;
}
while((fd2 = read(fd1, buffer, sizeof(buffer))) > 0)
write(1, buffer, fd2);
close(fd1);
}
int main(int argc, char *argv[])
{
int an = 1;
typefile(argv[an]);
return 0;
}
```

Teacher's Signature