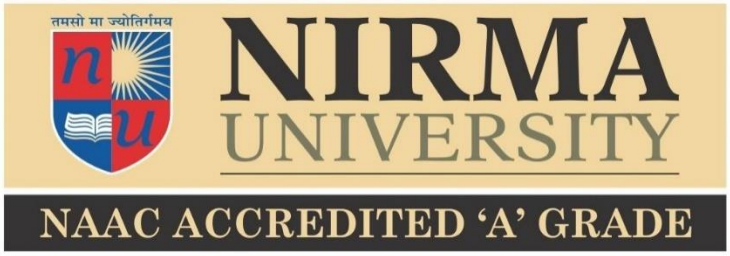

Name: Swekit Patel

Roll No: 22BCE347

Batch: F1

Subject: Operating System (OS)

Subject Code: 2CS506

Practical 9

9A

Code:

```
#include <stdio.h>
#include <unistd.h>
void one_fork()
{
 printf("Printing one fork function\n");
 int id;
 id = fork();
 if (id < 0)
 {
 printf("Fork failed\n");
 return;
 }
 else if (id == 0)
 {
 printf("Child process: fork() returned %d\n"
, id);
 }
 else
 {
 printf("Parent process: fork() returned %d\n"
, id);
 printf("Parent process: Child process finished.\n");
 }
}
void two_fork()
{
 printf("Printing two fork function\n");
 int id1, id2;
 id1 = fork();
 id2 = fork();
 if (id1 < 0 || id2 < 0)
 {
 printf("Fork failed\n");
 return;
 }
 else if (id1 == 0 && id2 == 0)
 printf("Child process 1: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id1 == 0)
 printf("Child process 2: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id2 == 0)
```

```
 printf("Child process 3: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else
 printf("Parent process: My PID is %d\n"
, getpid());
}
void three_fork()
{
 printf("Printing three fork function\n");
 int id1, id2, id3;
 id1 = fork();
 id2 = fork();
 id3 = fork();
 if (id1 < 0 || id2 < 0 || id3 < 0)
 {
 printf("Fork failed\n");
 return;
 }
 else if (id1 == 0 && id2 == 0 && id3 ==
0)
 printf("Child process 1: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id1 == 0 && id2 > 0 && id3 > 0)
 printf("Child process 2: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id1 > 0 && id2 == 0 && id3 > 0)
 printf("Child process 3: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id1 > 0 && id2 > 0 && id3 == 0)
 printf("Child process 4: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id1 == 0 && id2 == 0 && id3 >
0)
 printf("Child process 5: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id1 > 0 && id2 == 0 && id3 ==
0)
 printf("Child process 6: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else if (id1 == 0 && id2 > 0 && id3 ==
0)
```

```
 printf("Child process 7: My PID is %d, my parent's PID is %d\n"
, getpid(),
getppid());
 else
 printf("Parent process: My PID is %d\n"
, getpid());
}
int main()
{
 int n;
 printf("Enter the number of fork fuction (1,2,3): ");
 scanf("%d"
, &n);
 switch (n)
 {
 case 1:
 one_fork();
 break;
 case 2:
 two_fork();
 break;
 case 3:
 three_fork();
 break;
 default:
 printf("Invalid input\n");
 break;
 }
 return 0;
}
```

Output

```
Enter the number of fork fuction (1,2,3): 3
Printing three fork function
Parent process: My PID is 69038
Child process 4: My PID is 69043, my parent's PID is 1
Child process 3: My PID is 69042, my parent's PID is 1
Child process 2: My PID is 69041, my parent's PID is 1
Child process 6: My PID is 69045, my parent's PID is 1
Child process 7: My PID is 69046, my parent's PID is 1
Child process 5: My PID is 69044, my parent's PID is 1
Child process 1: My PID is 69047, my parent's PID is 69044
```

9B

Code:

```c
#include <stdio.h>
#include <string.h>
int main() {

 FILE *file;
 char filename[] = "hello.txt";
 char line[256];
 char word[100];
 printf("Enter the word to find:");
 scanf("%s"
,&word);
 file = fopen(filename, "r");
 if (file == NULL) {
 printf("\nError opening file %s",filename);
 return 1;
 }
 int i=1;
 int count=0;
 while (fgets(line, sizeof(line), file)){
 if (strstr(line, word) != NULL) {
 printf("\n%d | %s"
,i,line);
 count++;
 }
 i++;
 }
 printf("\nThe count of occurences are %d\n",count);
 fclose(file);
 return 0;
}
```

Output

```
PS C:\Users\SWEKIT\Documents\Code\OS Practical>  & 'c:\Users\SWEKIT\.vscode\extensions\ms-vscode.cpptools-1.19
owsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-rk4ucz12.0b3' '--stdout=Microsoft-MIEngine-Out-5ls0oobq.t
ror-5bg4abm3.zii' '--pid=Microsoft-MIEngine-Pid-35nwmknu.gyn' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--inter
Enter the word to find:swekit

1 | swekit
The count of occurences are 1
PS C:\Users\SWEKIT\Documents\Code\OS Practical>
```