Name: Ashmit Thawait

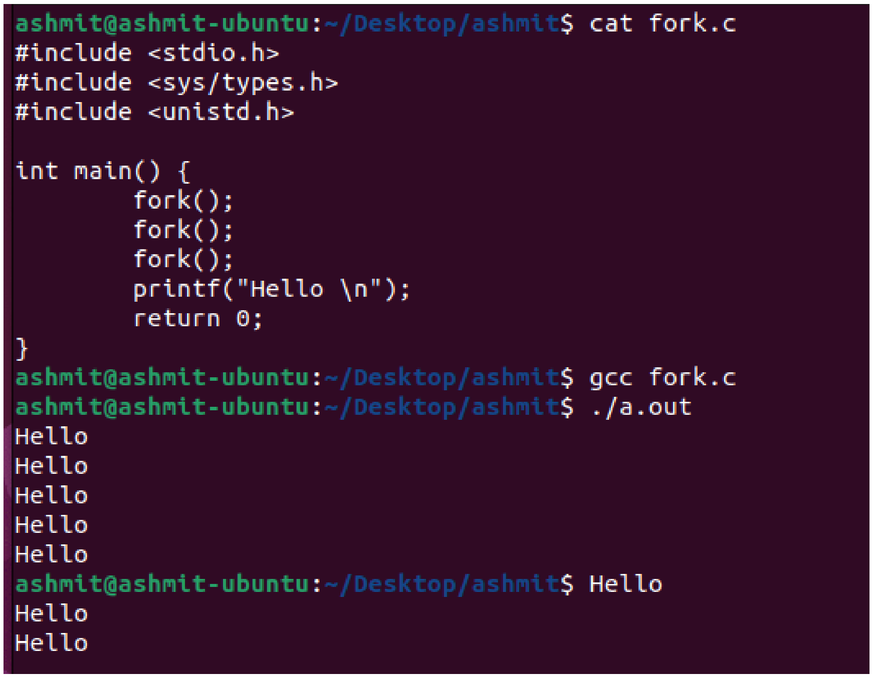
Roll No: 102203790

2CO-17

OS Assignment 4

Q1 : Write a program to implement fork () system call.

A1 :

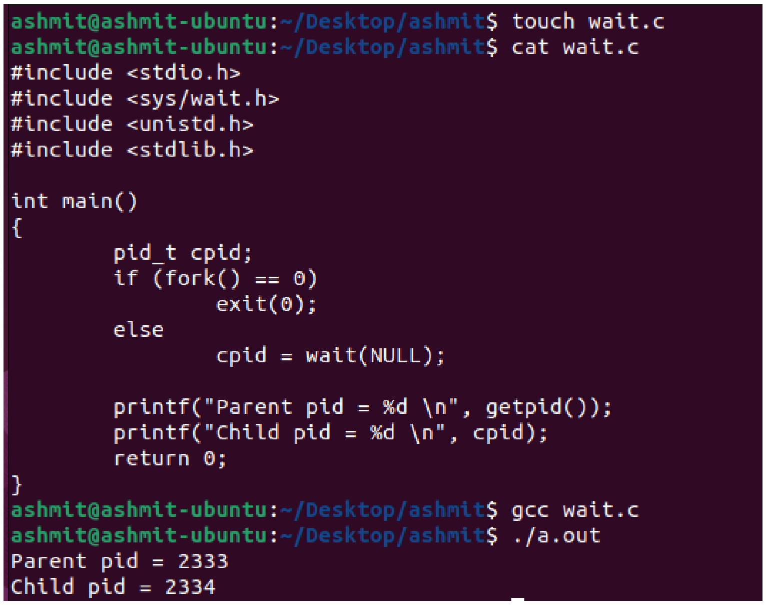


```
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ cat fork.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main() {
        fork();
        fork();
        fork();
        printf("Hello \n");
        return 0;
}
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ gcc fork.c
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ ./a.out
Hello
Hello
Hello
Hello
Hello
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ Hello
Hello
Hello
```

Q2 : Write a program to implement wait () and exit () System Calls.

A2 :



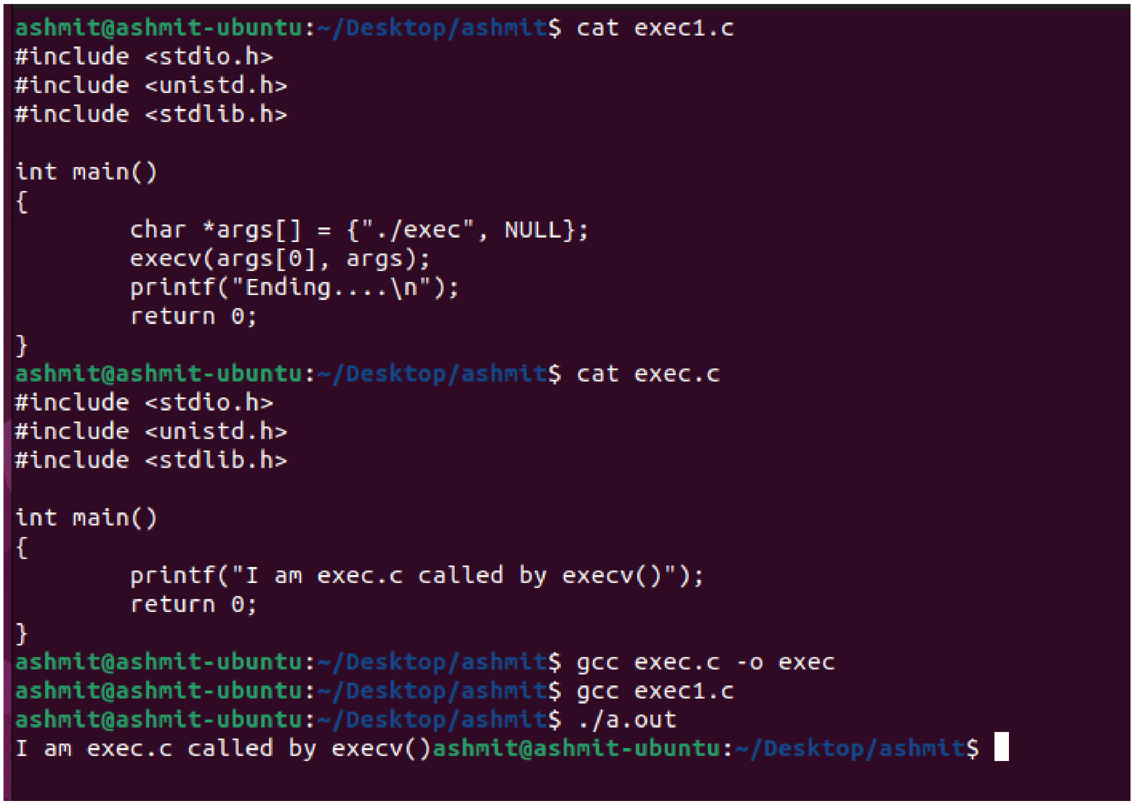
```
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ touch wait.c
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ cat wait.c
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
        pid_t cpid;
        if (fork() == 0)
                exit(0);
        else
                cpid = wait(NULL);

        printf("Parent pid = %d \n", getpid());
        printf("Child pid = %d \n", cpid);
        return 0;
}
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ gcc wait.c
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ ./a.out
Parent pid = 2333
Child pid = 2334
```

Q3 : Write a program to implement execv() system call.

A3 :



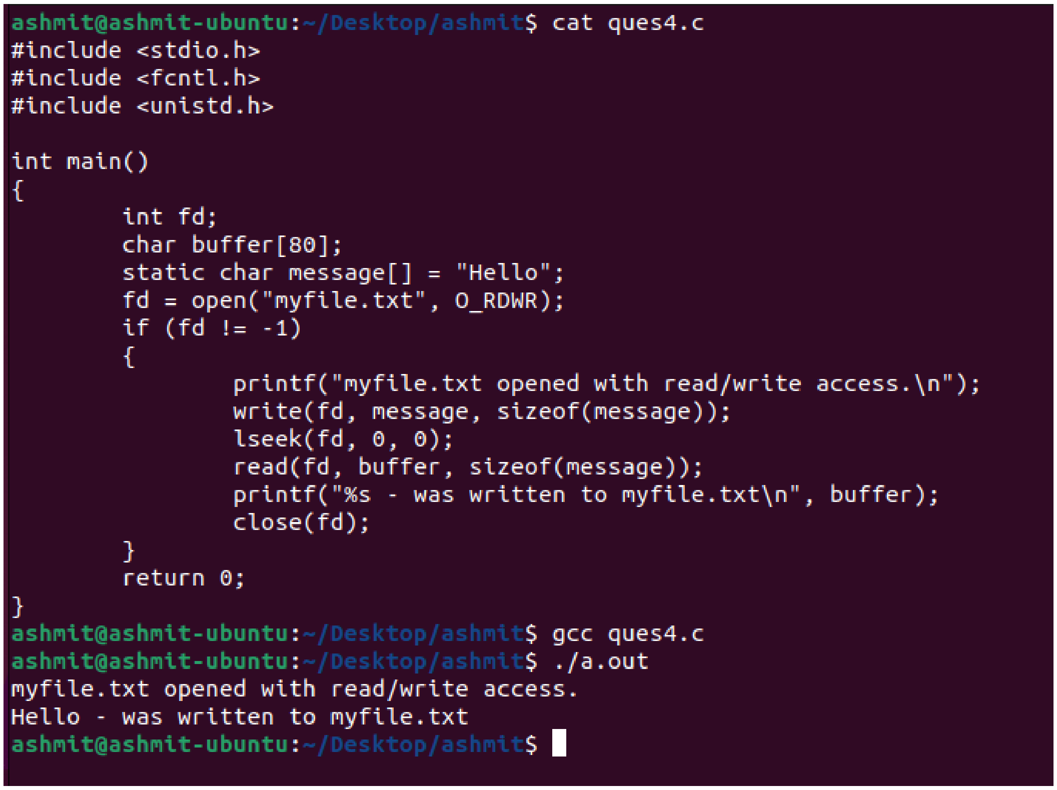
```
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ cat exec1.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
        char *args[] = {"./exec", NULL};
        execv(args[0], args);
        printf("Ending....\n");
        return 0;
}
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ cat exec.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
        printf("I am exec.c called by execv()");
        return 0;
}
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ gcc exec.c -o exec
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ gcc exec1.c
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ ./a.out
I am exec.c called by execv()ashmit@ashmit-ubuntu:~/Desktop/ashmit$ 
```

Q4 : Write a program to implement the system calls open (), read (), write () & close ().

A4 :



```
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ cat ques4.c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>

int main()
{
        int fd;
        char buffer[80];
        static char message[] = "Hello";
        fd = open("myfile.txt", O_RDWR);
        if (fd != -1)
        {
                printf("myfile.txt opened with read/write access.\n");
                write(fd, message, sizeof(message));
                lseek(fd, 0, 0);
                read(fd, buffer, sizeof(message));
                printf("%s - was written to myfile.txt\n", buffer);
                close(fd);
        }
        return 0;
}
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ gcc ques4.c
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ ./a.out
myfile.txt opened with read/write access.
Hello - was written to myfile.txt
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ 
```