

**U18CO018**  
**Shubham Shekhaliya**  
**Assignment – 1 (SS)**

Fork()

Code:-

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
    fork();

    printf("Hello world!\n");
    return 0;
}
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ g++ fork.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ ./a.out
Hello world!
Hello world!
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$
```

Exec()

Code:-

Execmain.c

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main() {
    char *args[]={"./EXEC",NULL};
    execvp(args[0],args);

    printf("Ending-----");

    return 0;
}
```

## Exec.c

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main() {
    char *args[]={"/EXEC",NULL};
    execvp(args[0],args);

    printf("Ending-----");

    return 0;
}
```

## Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  2: Code
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ gcc EXEC.c -o EXEC
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ g++ execmain.c
execmain.c: In function 'int main()':
execmain.c:5:23: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
   5 |         char *args[]={"/EXEC",NULL};
      |         ~~~~~^~~~~~
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ ./a.out
I am EXEC.c called by execvp()
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$
```

## Getpid()

### Code:-

```
#include<stdio.h>
#include<unistd.h>
int main() {
    printf("Current Process id %d\n" , getpid());
    return 0;
}
```

## Output:-

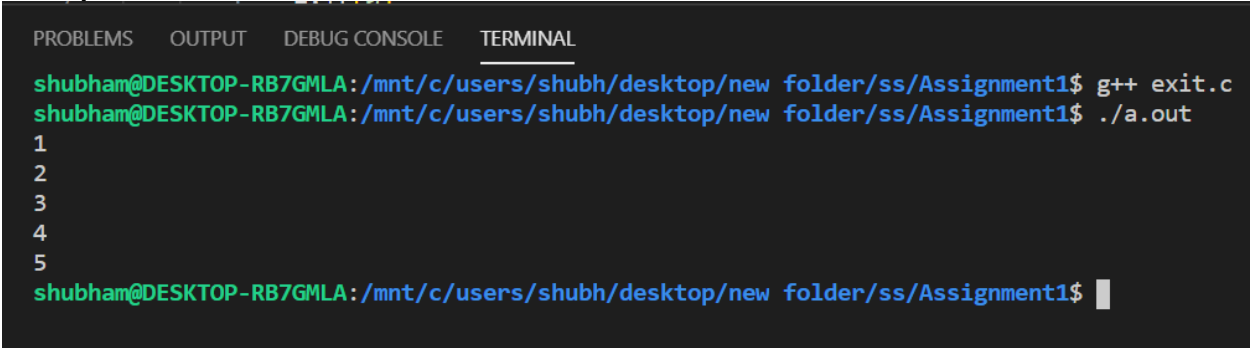
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ g++ getpid.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ ./a.out
Current Process id 8242
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$
```

exit()

Code:-

```
#include<stdio.h>
#include<stdlib.h>
int main() {
    for (int i = 1;i<=10;i++) {
        printf("%d\n",i);
        if(i==5) {
            exit(0);
        }
    }
    return 0;
}
```

Output:-



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ g++ exit.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$ ./a.out
1
2
3
4
5
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/Assignment1$
```

Open(), read(), write(), close(), stat()

Code:-

```
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>
int main() {
    int fd1 = open("./input.txt", O_RDONLY);
    int fd2 = open("./output.txt", O_WRONLY);
    if (fd1 < 0 || fd2 < 0) {
        printf("Cannot open files\n");
        exit(1);
    }
    struct stat st;
```

```

    fstat(fd1, &st);
    int size = st.st_size;
    char *c = (char *)calloc(size, sizeof(char));
    read(fd1, c, size);
    printf("%s", c);
    write(fd2, c, size);
    close(fd1);
    close(fd2);
    return 0;
}

```

Output:-

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ orwcs.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
Hiii
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ █

```

Opendir() closedir() readdir()

Code:-

```

#include <dirent.h>
#include <stdio.h>
#include <unistd.h>
int main(void) {
    struct dirent *de;
    DIR *dr = opendir(".");
    if (dr == NULL) {
        printf("Could not open current directory");
        _exit(1);
    }
    while ((de = readdir(dr)) != NULL) {
        printf("%s\n", de->d_name);
    }
    closedir(dr);
    return 0;
}

```

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
.
..
a.out
alarm.c
brksbrk.c
chmod.c
EXEC
exec.c
execmain.c
exit.c
fork.c
getpid.c
input.txt
kill.c
lseek.c
mmap.c
munmap.c
ocr.c
orw.c
output.txt
pause.c
pipe.c
preadpwrite.c
readvwrite.c
rt_sigaction.c
```

## Chmod()

Code:-

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/stat.h>
int main() {
    mode_t mode = 0755;
    uid_t owner = 01000;
    uid_t group = 01000;
    chmod("./output.txt", mode);
    chown("./output.txt", owner, group);
    return 0;
}
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: bash

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ chmod.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ chmod 444 output.txt
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ls -lh output.txt
-r-xr-xr-x 1 shubham shubham 19 Jan 31 23:01 output.txt
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ls -lh output.txt
-rwxrwxrwx 1 shubham shubham 19 Jan 31 23:01 output.txt
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$
```

## lseek()

Code:-

```
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
    int fd = open("input.txt", O_RDONLY);
    struct stat st;
    fstat(fd, &st);
    int size = st.st_size;
    char *c1 = (char *)calloc(size, sizeof(char));
    char *c2 = (char *)calloc(size, sizeof(char));
    read(fd, c1, size);
    printf("%s\n", c1);
    // Move back to beginning of file
    lseek(fd, 0, SEEK_SET);
    read(fd, c2, size);
    printf("%s", c2);
    close(fd);
    return 0;
}
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
```

```
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ lseek.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
Hiiii
How are You?

Hiiii
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$
```

mmap(), munmap()

Code:-

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/mman.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
    int fd = open("./input.txt", O_RDONLY);
    struct stat st;
    fstat(fd, &st);
    int size = st.st_size;
    char *data = (char *)mmap(NULL, size, PROT_READ, MAP_PRIVATE, fd, 0);
    printf("%s", data);
    munmap(data, size);
    return 0;
}
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
```

```
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ mmunmap.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
Hiiii
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$
```

brk(), sbrk()

Code:-

```
#include <unistd.h>
int main() {
    int *p = (int *)sbrk(0);
    //brk (p + 4);
    *p = 1;
    return 0;
}
```

Output:-

Without brk()

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ brksbrk.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$
```

With brk()

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL 1: ba

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ brksbrk.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$
```

rt\_sigaction()

Code:-

```
#include <signal.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
void handler(int num) {
    char *buf = "Ctrl C Called! \n";
    int len = strlen(buf);
    write(STDOUT_FILENO, buf, len);
}
int main() {
    struct sigaction sa;
    sa.sa_handler = handler;
    sigaction(SIGINT, &sa, NULL);
    printf("PID: %d", getpid());
    while (1)
```



```

{
    printf("...");
    fflush(stdout);
    sleep(1);
}
return 0;
}

```

## Output:-

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: bash

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ rt_signaction.c
rt_signaction.c: In function 'void handler(int)':
rt_signaction.c:6:17: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
   6 |     char *buf = "Ctrl C Called! \n";
     |                   ^~~~~~

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
PID: 12922.....
.....^C
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$

```

It should not stop after presses ctrl+c but on my pc not working.

pread(), pwrite()

Code:-

```

#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>
#define OFFSET 4
int main() {
    int fd1 = open("./input.txt", O_RDONLY); //input
    int fd2 = open("./output.txt", O_WRONLY); //output
    if (fd1 < 0 || fd2 < 0)
    {
        printf("Cannot open files\n");
        exit(1);
    }
    struct stat st;
    fstat(fd1, &st);
    int size = st.st_size;
    char *c = (char *)calloc(size, sizeof(char));
    pread(fd1, c, size, OFFSET);
    printf("%s", c);
    pwrite(fd2, c, size - OFFSET, 0);
    close(fd1);
}

```

```

    close(fd2);
    free(c);
    return 0;
}

```

## Output:-

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: bash

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ preadpwrite.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
i
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ cat input.txt
Hiiii
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ cat output.txt
i
How are You?
ou?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$

```

## readv(), writev()

### Code:-

```

#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <unistd.h>
#define VECTOR_COUNT 3
int main() {
    int fd1 = open("input.txt", O_RDONLY); //input
    int fd2 = open("output.txt", O_WRONLY); //output
    struct iovec vec[VECTOR_COUNT];
    struct stat st;
    fstat(fd1, &st);
    int size = st.st_size;
    for (int i = 0; i < VECTOR_COUNT - 1; ++i)
    {
        vec[i].iov_base = (char *)calloc(size / VECTOR_COUNT, sizeof(char));
        vec[i].iov_len = size / VECTOR_COUNT;
    }
    int rem = (size - (size / VECTOR_COUNT) * (VECTOR_COUNT - 1));
    vec[VECTOR_COUNT - 1].iov_base = (char *)calloc(rem, sizeof(char));
    vec[VECTOR_COUNT - 1].iov_len = rem;
}

```

```

    readv(fd1, vec, VECTOR_COUNT);
    writev(fd2, vec, VECTOR_COUNT);
    for (int i = 0; i < VECTOR_COUNT; ++i)
    {
        printf("%s", (char *)vec[i].iov_base);
    }
    return 0;
}

```

## Output:-

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: bash

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ readvwritev.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
Hiiii
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ cat input.txt
Hiiii
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ cat output.txt
Hiiii
How are You?
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ 

```

## alarm()

### Code:-

```

#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <unistd.h>
void sig_handler(int signum) {
    printf("Alarm Called\n");
    exit(0);
}
int main() {
    signal(SIGALRM, sig_handler);
    alarm(2);
    while (1)
    {
        printf("...");
        fflush(stdout);
        sleep(1);
    }
    return 0;
}

```

## Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ alarm.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
.....Alarm Called
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$
```

## kill()

### Code:-

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <sys/types.h>
int main(){
    int pid = getpid(), x = 1;
    printf("PID: %d\n", pid);
    while (x++)
    {
        printf("Running...\n");
        sleep(1);
        if (x == 5)
            kill(pid, SIGINT);
    }
    return 0;
}
```

## Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ kill.c
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
PID: 12571
Running...
Running...
Running...
Running...

shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$
```

## pipe()

Code:-

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define OPSIZE 16
char *op1 = "output 1";
char *op2 = "output 2";
char *op3 = "output 3";
int main() {
    char inbuf[OPSIZE];
    int p[2], i;
    if (pipe(p) < 0)
        exit(EXIT_FAILURE);
    write(p[1], op1, OPSIZE);
    write(p[1], op2, OPSIZE);
    write(p[1], op3, OPSIZE);
    for (i = 0; i < 3; i++)
    {
        read(p[0], inbuf, OPSIZE);
        printf("%s\n", inbuf);
    }
    return 0;
}
```

Output:-

```
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new_folder/ss/assignment1$ ./a.out
output 1
output 2
output 3
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new_folder/ss/assignment1$
```

## pause()

Code:-

```
#include <stdio.h>
#include <unistd.h>
int main(void) {
    for (int i = 0; i < 10; i++) {
        printf("%d\n", i);
        if (i == 5)
            pause();
    }
    return 0;
}
```

## Output:-

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ g++ pause.c
```

```
shubham@DESKTOP-RB7GMLA:/mnt/c/users/shubh/desktop/new folder/ss/assignment1$ ./a.out
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
█
```