**Linux Programming**

**lab-8**

***G Nanda kishore Reddy***

**17MIS1066**

1. **Fork**

Code:

#include<stdio.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

int pid;

pid=fork();

if(pid==0)

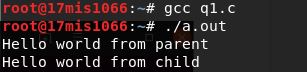
printf("Hello world from child\n");

else

printf("Hello world from parent\n");

}

Output:



2. **EXECCVE**

**File Name: execve.c**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

int main()

{

int a=fork();

if(a==0)

execve("ex2",NULL,NULL); //pass ex2.c object here

}

**File Name: ex2.c**

#include<stdio.h>

int main()

{

int i,a;

int m=0;

for(i=1;i<=5;i++)

{

m=m+i;

}

printf("%d\n",m);

}

Output:



**3. Executing cat command using execve:**

**CODE:**

#include<stdio.h>

#include<stdlib.h>

int main()

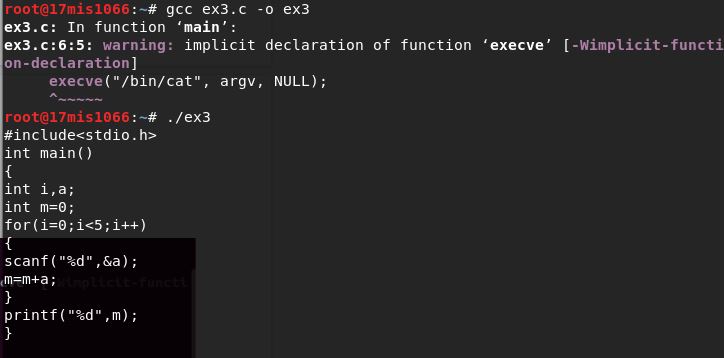
{

char\* argv[3] = {"/bin/cat", "ex2.c", NULL};

execve("/bin/cat", argv, NULL);

}

Output:



4.

Code:

#include<stdio.h>

#include<stdlib.h>

#include<sys/wait.h>

int main(void)

{

pid\_t pid;

int status;

if((pid=fork())<0)

{

printf("Error");

}

else if(pid==0)

{

exit(0);

}

if(wait(&status!=pid))

{

printf("wait Error\n");

}

//pr\_exit(status);

if((pid=fork())<0)

{

printf("Fork Error");

}

else if(pid==0)

{

abort();

}

}

Output:

