

## OS EXPERIMENT NO : 3

Name: Das Rahul Sanjay

SE-9

Roll No: 10

AIM : Create a child process in Linux using the fork system call. From the child process obtain the process ID of both child and parent by using getpid and getppid system call. Explore wait, getpid, getuid , getgid getegid, geteuid and waitpid before termination of process


CODE :

```
GNU nano 4.9.3 exp3.c Modified
#include<stdio.h>
#include<unistd.h>
void main()
{
int pid,ppid,n=4,key=7,i,a[4]={1,2,3,7};
ppid=getppid();
```

```
pid=fork();
if(pid<0)
{
printf("\n\n");
printf("Error Creating Child Process\n");
}
else if (pid==0)
{
printf("\n\n");
printf("Process id:\t%d\n",getpid());
printf("Parent Process id:\t%d\n",getppid());
printf("PPid :\t%d\n",ppid);
}
else
{
printf("\n\n\nLinear Search Program \n\n\n");
for(i=0;i<n;i++)
{
if(a[i]==key)
```

```
GNU nano 4.9.3      exp3.c      Modified
pid=fork();
if(pid<0)
{
printf("\n\n");
printf("Error Creating Child Process\n");
}
else if (pid==0)
{
printf("\n\n");
printf("Process id:\t%d\n",grtpaid());
printf("Parent Process id:\t%d\n",getppid());
printf("PPid :\t%d\n",ppid);
}
else
{
printf("\n\n\nLinear Search Program \n\n\n");
for(i=0;i<n;i++)
{
if(a[i]==key)
{
printf("Element %d found at %d position\n",key,i+i);
}
}
}
}
```

**^G** Get Help   **^O** Write Out   **^W** Where Is   **^K** Cut Text   **^J** Justify   **^C** Cur Pos  
**^X** Exit   **^R** Read File   **^\_\** Replace   **^U** Paste Text   **^T** To Spell   **^\_** Go To Line



```
for(i=0;i<n;i++)
{
if(a[i]==key)
{
printf("Element %d found at %d position\n",key,i+i);
}
}
}
```

OUTPUT :

```
Process id:    180
Parent Process id:    179
PPid : 62

Element 7 found at 6 position
localhost:~#
```