



**NAME : SUFYAN AHMAD**

**SAP ID : 55766**

**SECTION : SE 5-2**

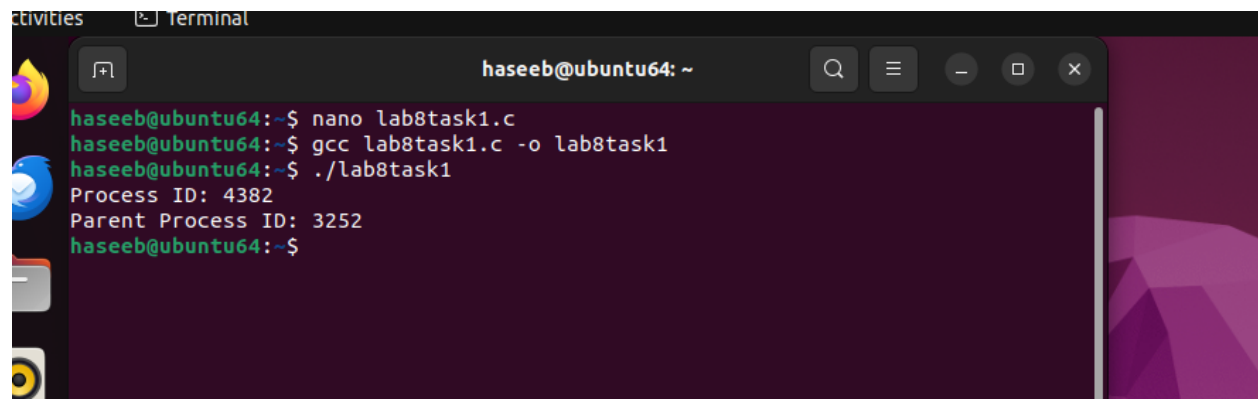
**LAB TASK : 08**

---

**TASK : 01**

**ANSWER :**

**STEP : 01**

A terminal window titled 'haseeb@ubuntu64: ~' showing the execution of a C program. The user runs 'nano lab8task1.c', then 'gcc lab8task1.c -o lab8task1', and finally './lab8task1'. The output shows 'Process ID: 4382' and 'Parent Process ID: 3252'.

```
haseeb@ubuntu64:~$ nano lab8task1.c
haseeb@ubuntu64:~$ gcc lab8task1.c -o lab8task1
haseeb@ubuntu64:~$ ./lab8task1
Process ID: 4382
Parent Process ID: 3252
haseeb@ubuntu64:~$
```

STEP :02

A screenshot of the nano text editor showing the source code of lab8task1.c. The code includes <stdio.h> and <unistd.h>, and defines a main function that prints the process ID and parent process ID using getpid() and getppid().

```
GNU nano 6.2 lab8task1.c
#include <stdio.h>
#include <unistd.h>

int main() {
    printf("Process ID: %d\n", getpid());
    printf("Parent Process ID: %d\n", getppid());
    return 0;
}
```

TASK : 02

ANSWER :

STEP : 01

```
haseeb@ubuntu64: ~  
GNU nano 6.2 lab8_e2.c *  
#include <unistd.h>  
#include <stdio.h>  
#include <stdlib.h>  
  
int main(void) {  
    fork();  
    int x = 5;  
    pid_t pid = getpid();  
    printf("Value of X in PID = %d is %d\n", pid, x);  
    return 0;  
}
```

STEP : 02

```
haseeb@ubuntu64:~$ nano lab8_e2.c  
haseeb@ubuntu64:~$ gcc lab8_e2.c -o lab8_e2  
haseeb@ubuntu64:~$ ./lab8_e2  
Value of X in PID = 4421 is 5  
Value of X in PID = 4420 is 5  
haseeb@ubuntu64:~$
```

TASK : 03

ANSWER:

STEP : 01

```
haseeb@ubuntu64: ~  
GNU nano 6.2 lab8_e3.c *  
#include <unistd.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/types.h>  
  
int main(void) {  
    fork();  
    pid_t pid = getpid();  
    int i;  
    for (i = 1; i <= 200; i++) {  
        printf("This line is from PID %d, value = %d\n", pid, i);  
    }  
    return 0;  
}
```

## STEP : 02

```
haseeb@ubuntu64: ~  
This line is from PID 4453, value = 178  
This line is from PID 4453, value = 179  
This line is from PID 4453, value = 180  
This line is from PID 4453, value = 181  
This line is from PID 4453, value = 182  
This line is from PID 4453, value = 183  
This line is from PID 4453, value = 184  
This line is from PID 4453, value = 185  
This line is from PID 4453, value = 186  
This line is from PID 4453, value = 187  
This line is from PID 4453, value = 188  
This line is from PID 4453, value = 189  
This line is from PID 4453, value = 190  
This line is from PID 4453, value = 191  
This line is from PID 4453, value = 192  
This line is from PID 4453, value = 193  
This line is from PID 4453, value = 194  
This line is from PID 4453, value = 195  
This line is from PID 4453, value = 196  
This line is from PID 4453, value = 197  
This line is from PID 4453, value = 198  
This line is from PID 4453, value = 199  
This line is from PID 4453, value = 200  
haseeb@ubuntu64:~$
```

TASK : 04

ANSWER :

STEP : 01

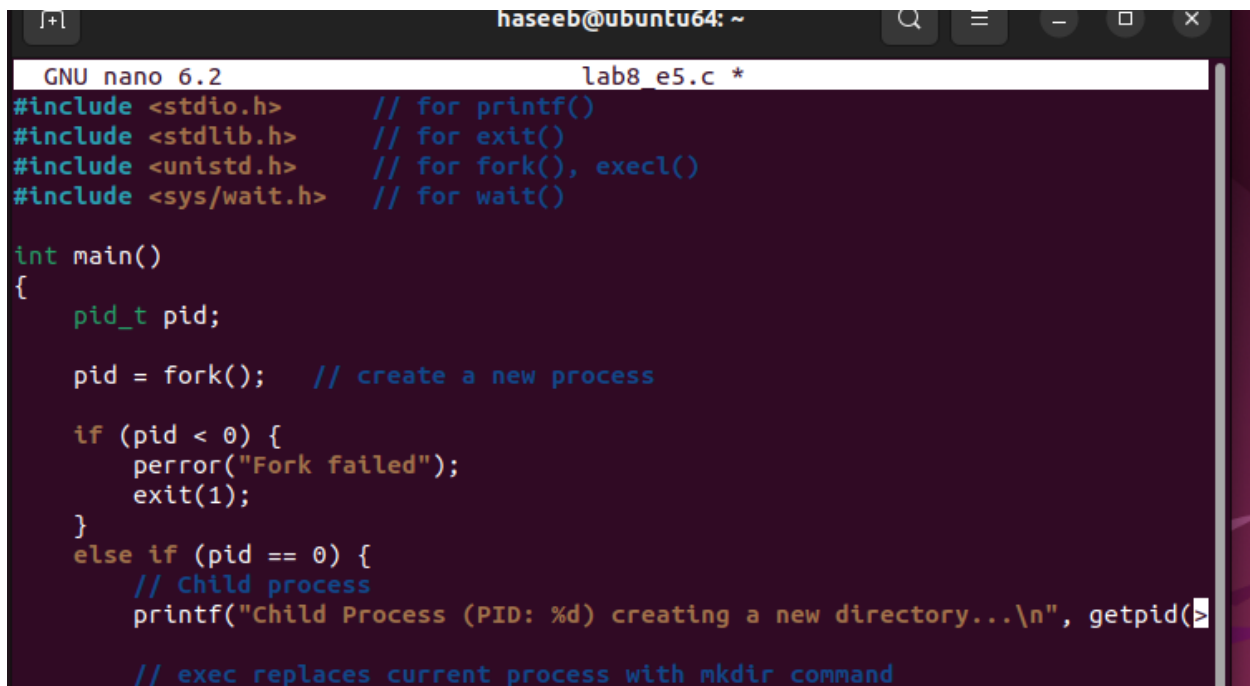
```
haseeb@ubuntu64:~$ nano lab8_e4.c
haseeb@ubuntu64:~$ gcc lab8_e4.c -o lab8_e4
haseeb@ubuntu64:~$ ./lab8_e4
This is the Child Process (PID: 4874)
Child executing command: ls -l
Parent Process (PID: 4873) waiting for child...
total 168
```

---

TASK : 05

ANSWER :

STEP : 01



```
haseeb@ubuntu64: ~
GNU nano 6.2 lab8_e5.c *
#include <stdio.h> // for printf()
#include <stdlib.h> // for exit()
#include <unistd.h> // for fork(), execl()
#include <sys/wait.h> // for wait()

int main()
{
    pid_t pid;

    pid = fork(); // create a new process

    if (pid < 0) {
        perror("Fork failed");
        exit(1);
    }
    else if (pid == 0) {
        // Child process
        printf("Child Process (PID: %d) creating a new directory...\n", getpid());

        // exec replaces current process with mkdir command
    }
}
```

STEP 02 :

```
haseeb@ubuntu64:~$ nano lab8_e5
haseeb@ubuntu64:~$ nano lab8_e5.c
haseeb@ubuntu64:~$ gcc lab8_e5.c -o lab8_e5
haseeb@ubuntu64:~$ ./lab8_e5
Child Process (PID: 4921) creating a new directory...
Parent Process (PID: 4920) waiting for child...
mkdir: cannot create directory 'newdir': File exists
Parent Process: Child finished.
haseeb@ubuntu64:~$
```