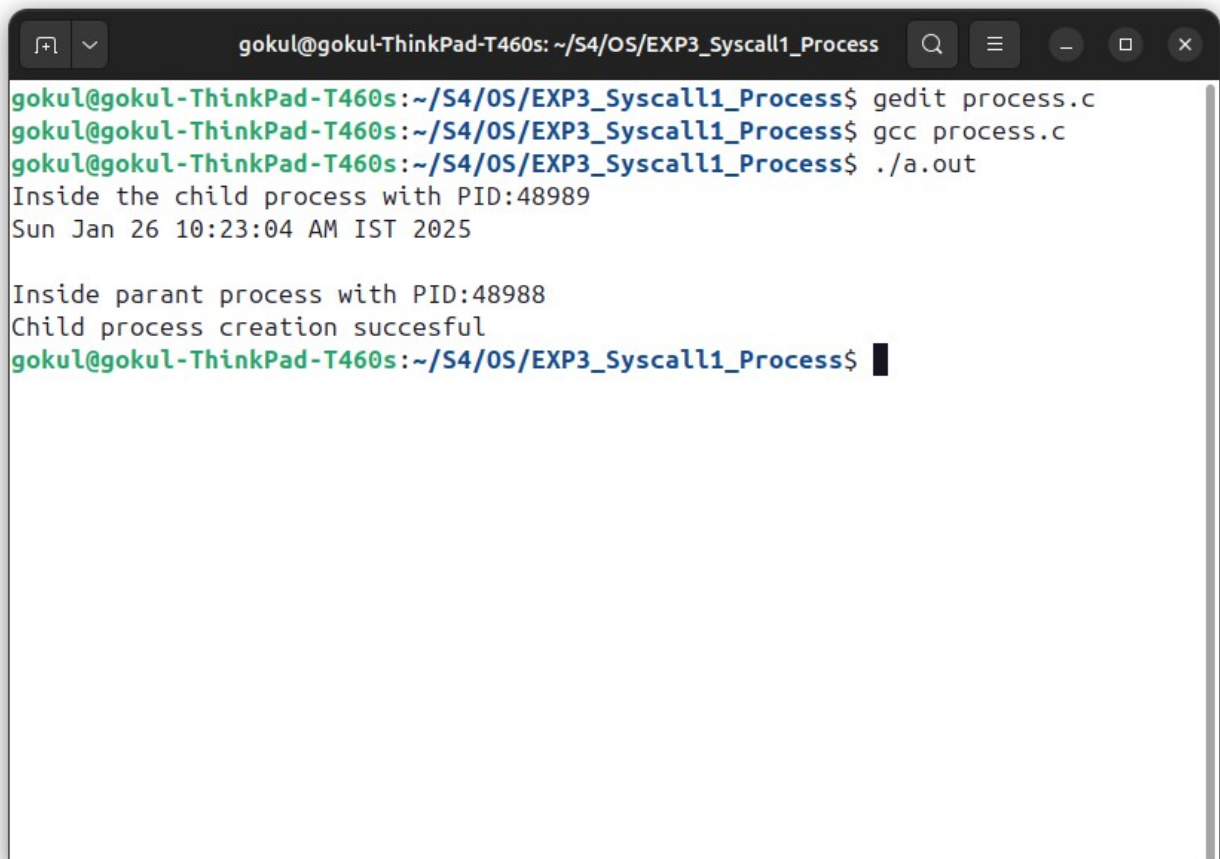


```

#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<sys/wait.h>

int main(){
    int status,pid,child_pid;
    pid=fork();
    if(pid==-1)
    {
        printf("Child process creation failed\n");
        return -1;
    }
    else if(pid==0){
        printf("Inside the child process with PID:%d\n",getpid());
        execlp("/bin/date","date",NULL);
        return 0;
    }
    else{
        child_pid=wait(&status);
        printf("\nInside parant process with PID:%d\n",getpid());
        printf("Child process creation succesful\n");
        return 0;
    }
}

```



The image shows a terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP3_Syscall1_Process'. The user has executed the following commands: 'gedit process.c', 'gcc process.c', and './a.out'. The output of the program is displayed in the terminal, showing the child process running 'date' and the parent process waiting for it to complete.

```

gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$ gedit process.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$ gcc process.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$ ./a.out
Inside the child process with PID:48989
Sun Jan 26 10:23:04 AM IST 2025

Inside parant process with PID:48988
Child process creation succesful
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$

```

