

C task1.c ×

Lab 4 > C task1.c > ...

```
1  #include<stdio.h>
2  #include<sys/types.h>
3  #include<unistd.h>
4
5  int main(){
6
7      fork();
8      printf("Hello World\n");
9      fork();
10
11 }
```

baigu pop-os ~/Desktop/OS Lab ./obj

Hello World

Hello World

C task2.c ×

Lab 4 > C task2.c > ...

```
1  #include<stdio.h>
2  #include<unistd.h>
3  #include<sys/types.h>
4  #include<stdlib.h>
5
6  int main (void) {
7
8      pid_t pid=fork();
9      if( pid == 0 ) {
10         printf("I am Child process \n") ;
11     }
12     else if( pid > 0 ) {
13         printf("I am Parent process \n") ;
14     }
15     else if ( pid < 0 ) {
16         printf ("Error in Fork");
17     }
18 }
```

baigu pop-os ~/Desktop/OS Lab ./obj

I am Parent process
I am Child process

```
run Terminal Help
C task3.c x
Lab 4 > C task3.c > ...
1  #include<stdio.h>
2  #include<unistd.h>
3  #include<sys/types.h>
4  #include<stdlib.h>
5  int main(){
6
7      int First,Second;
8      int HCF;
9      printf("Enter First Number & it must be positive: ");
10     scanf("%d", &First);
11     printf("\nEnter Second Number & it must be positive: ");
12     scanf("%d", &Second);
13
14     printf("First Number is : %d\n",First);
15     printf("Second Number is : %d\n",Second);
16
17     int pid = fork();
18
19     if(pid == 0){
20
21         printf("This is Child Process\n");
22
23         if(First > Second)
24             printf("First number is greater\n");
25         else
26             printf("Second number is greater\n");
27
28     }
29
30     else if(pid > 0){
31
32         printf("This is Parent Process\n");
33         for(int i=1; i<=First || i<=Second; i++)
34             if(First%i == 0 && Second%i == 0)
35                 HCF = i;
36
37         printf("HCF is = %d\n", HCF);
38
39     }
40
41     else
42         printf("Error");
43 }
```

```
baigu pop-os ~/Desktop/OS Lab ./obj
Enter First Number & it must be positive: 24

Enter Second Number & it must be positive: 12
First Number is : 24
Second Number is : 12
This is Parent Process
HCF is = 12
This is Child Process
First number is greater

baigu pop-os ~/Desktop/OS Lab
```

```

C task4.c x
Lab 4 > C task4.c > main()
8  int global = 0;
9  int main( ){
10
11     int status;
12     pid_t child_pid ;
13     int local = 0 ;
14
15     child_pid = fork ( ) ;
16     if ( child_pid >= 0 ) {
17
18         if ( child_pid == 0 )
19         {
20             printf ("child process!\n" ) ;
21
22             local++;
23             global ++;
24             printf ("child PID = %d , parent pid = %d\n" , getpid( ) , getppid( ) ) ;
25             printf ("\nchild's local = %d , child's global = %d\n" , local , global ) ;
26         }
27
28         else if(child_pid >0)
29         {
30
31             printf ("parent process!\n" ) ;
32             printf ("parent PID = %d , child pid = %d\n" , getpid( ) , child_pid ) ;
33             int w=wait(&status) ;
34             printf ( "\nParent's local = %d ,parent's global = %d\n" ,local , global ) ;
35             printf ( "Parent says bye!\n" ) ;
36         }
37
38         else
39         {
40             perror ("fork") ;
41             exit (0) ;
42         }
43     }
44     return 0;
45
46 }

```

```

baigu @ pop-os [ ~/Desktop/OS Lab ] ./obj

```

```

parent process!

```

```

parent PID = 7548 , child pid = 7549

```

```

child process!

```

```

child PID = 7549 , parent pid = 7548

```

```

child's local = 1 , child's global = 1

```

```

Parent's local = 0 ,parent's global = 0

```

```

Parent says bye!

```

```

baigu @ pop-os [ ~/Desktop/OS Lab ]

```