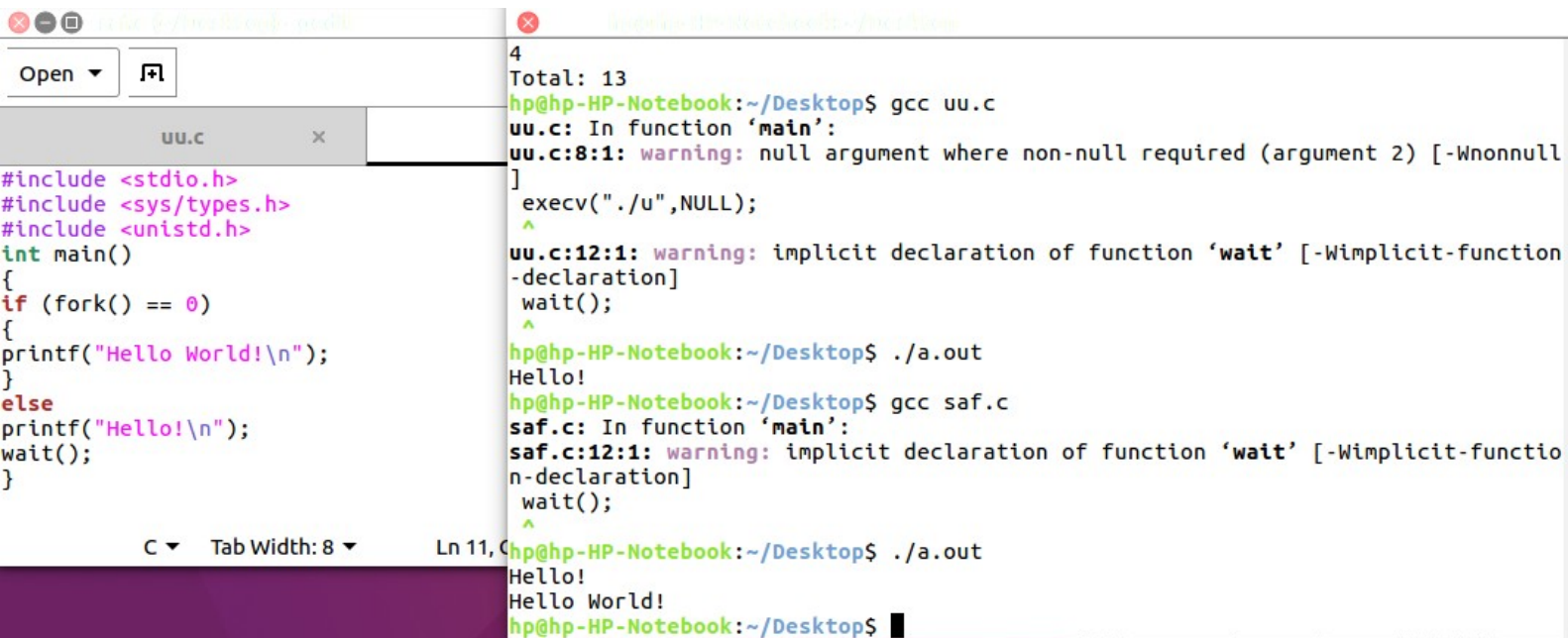


SWE4009- LINUX PROGRAMMING
LAB ASSIGNMENT -8
17MIS1028
ANJANA P

1.) SCREENSHOT:



The screenshot displays a Linux environment with a code editor on the left and a terminal on the right. The code editor shows the source code for `uu.c`, which includes `<stdio.h>`, `<sys/types.h>`, and `<unistd.h>`. The `main` function uses `fork()` to create a child process. The child process prints "Hello World!\n" and then calls `wait()`. The parent process prints "Hello!\n" and then calls `wait()`. The terminal shows the compilation of `uu.c` using `gcc`, which produces a warning about a null argument in `execv` and an implicit declaration of `wait`. The terminal also shows the execution of the resulting binary `a.out`, which outputs "Hello!" and "Hello World!".

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
if (fork() == 0)
{
printf("Hello World!\n");
}
else
printf("Hello!\n");
wait();
}
```

```
4
Total: 13
hp@hp-HP-Notebook:~/Desktop$ gcc uu.c
uu.c: In function 'main':
uu.c:8:1: warning: null argument where non-null required (argument 2) [-Wnonnull]
]
^
uu.c:12:1: warning: implicit declaration of function 'wait' [-Wimplicit-function
-declaration]
wait();
^
hp@hp-HP-Notebook:~/Desktop$ ./a.out
Hello!
hp@hp-HP-Notebook:~/Desktop$ gcc saf.c
saf.c: In function 'main':
saf.c:12:1: warning: implicit declaration of function 'wait' [-Wimplicit-functio
n-declaration]
wait();
^
hp@hp-HP-Notebook:~/Desktop$ ./a.out
Hello!
Hello World!
hp@hp-HP-Notebook:~/Desktop$
```

2.) SCREENSHOT:

The image shows a code editor window with two tabs: 'uu.c' and 'saf.c'. The 'saf.c' tab is active, displaying the following C code:

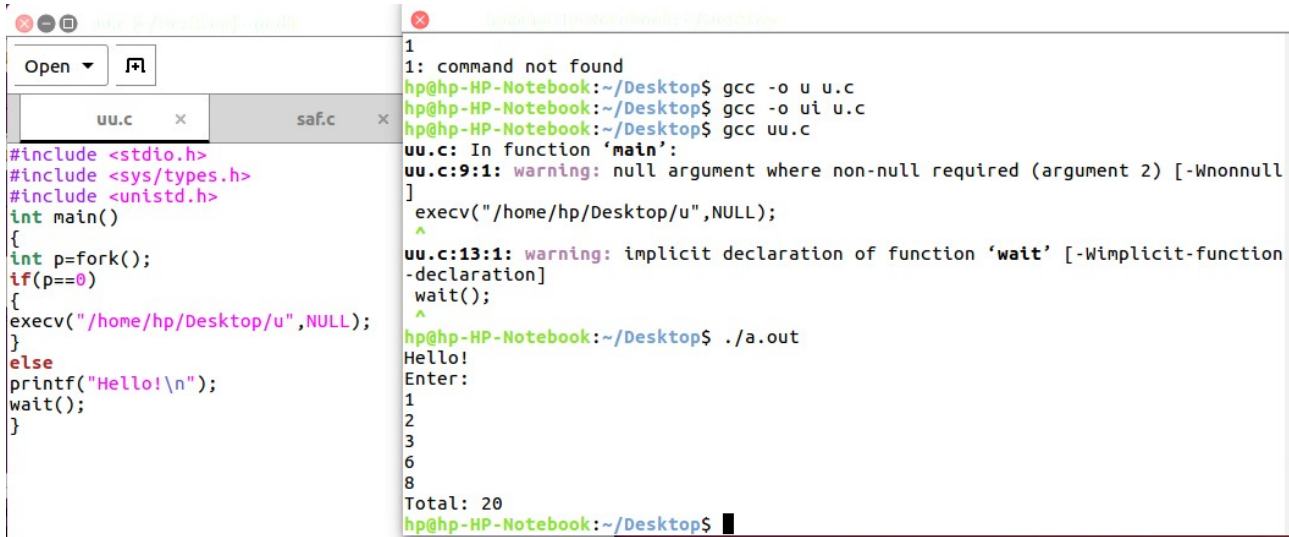
```
#include<sys/wait.h>
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<stdlib.h>
int main(void)
{
    pid_t pid;
    int status;
    if((pid = fork())<0)
    {
        printf("Fork error");
    }
    else if(pid ==0)
    {
        printf("Child\n");
        exit(0);}
    if (wait(&status)!=pid)
    {
        printf("wait error");
    }
    else
    {
        printf("Now Parent\n");
    }
    exit(status);
}
```

Below the code editor, the status bar shows 'C', 'Tab Width: 8', 'Ln 21, Col 2', and 'INS'.

Overlaid on the right side of the code editor is a terminal window. The terminal shows the following commands and output:

```
hp@hp-HP-Notebook:~$ cd Desktop
hp@hp-HP-Notebook:~/Desktop$ gcc saf.c
hp@hp-HP-Notebook:~/Desktop$ ./a.out
Child
Now Parent
hp@hp-HP-Notebook:~/Desktop$
```

3.) SCREENSHOT:



The screenshot shows a code editor on the left and a terminal window on the right. The code editor displays the source code for a C program named `uu.c`. The code includes `<stdio.h>`, `<sys/types.h>`, and `<unistd.h>`. It defines a `main` function that forks a child process. The child process calls `execv` to execute `/home/hp/Desktop/u` with `NULL` as arguments. The parent process then calls `wait` to wait for the child to finish. The parent process prints "Hello!\n" and then waits for user input. The terminal window shows the compilation of `uu.c` using `gcc` to produce `u` and `ui`. It also shows the execution of `uu.c` using `./a.out`, which outputs "Hello!" and then prompts the user to enter a number. The user enters 1, 2, 3, 6, and 8, and the program outputs "Total: 20".

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    int p=fork();
    if(p==0)
    {
        execv("/home/hp/Desktop/u",NULL);
    }
    else
    printf("Hello!\n");
    wait();
}
```

```
1
1: command not found
hp@hp-HP-Notebook:~/Desktop$ gcc -o u u.c
hp@hp-HP-Notebook:~/Desktop$ gcc -o ui u.c
hp@hp-HP-Notebook:~/Desktop$ gcc uu.c
uu.c: In function 'main':
uu.c:9:1: warning: null argument where non-null required (argument 2) [-Wnonnull]
]
    execv("/home/hp/Desktop/u",NULL);
    ^
uu.c:13:1: warning: implicit declaration of function 'wait' [-Wimplicit-function
-declaration]
    wait();
    ^
hp@hp-HP-Notebook:~/Desktop$ ./a.out
Hello!
Enter:
1
2
3
6
8
Total: 20
hp@hp-HP-Notebook:~/Desktop$
```