

## OS Lab Assignment

**Submitted By:**

Manroop Parmar

101906134 3EC6

1. write 'C' program to implement the process creation system calls.

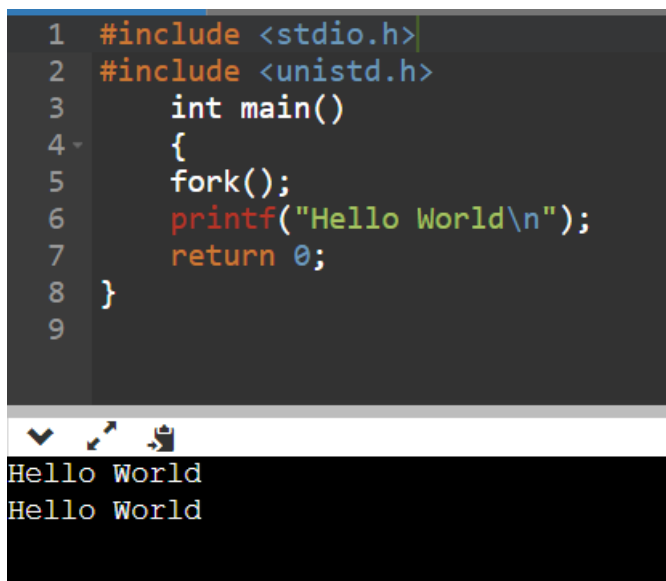
```
#include <stdio.h>

#include <unistd.h>

int main()
{
    fork();

    printf("Hello World");

    return 0;
}
```



```
1  #include <stdio.h>
2  #include <unistd.h>
3      int main()
4      {
5          fork();
6          printf("Hello World\n");
7          return 0;
8      }
9
```

✓ ↩ 📄

Hello World  
Hello World

2. write 'C' program to create child with 'sleep' command.

```
#include <stdio.h>

#include <unistd.h>

int main()
{
    sleep(5);

    fork();
}
```

```
printf("Hello\n");  
if(fork()==0)  
    printf("Child\n");  
else if(fork())>0  
    printf("Parent\n");  
else if(fork())<0  
    printf("Error\n");  
return 0;  
}
```

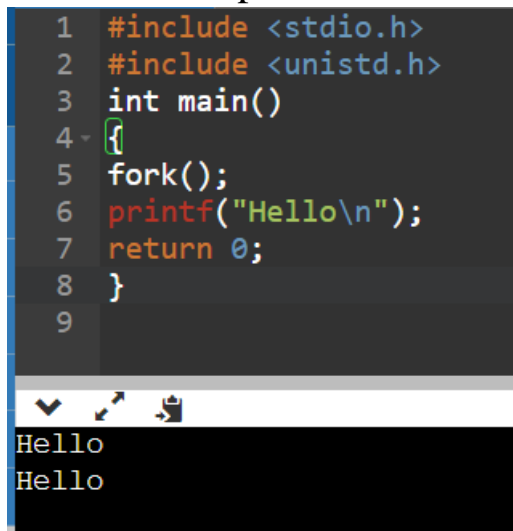
```
1  #include <stdio.h>  
2  #include <unistd.h>  
3  int main()  
4  {  
5      sleep(5);  
6      fork();  
7      printf("Hello\n");  
8      if(fork()==0)  
9          printf("Child\n");  
10     else if(fork())>0  
11         printf("Parent\n");  
12     else if(fork())<0  
13         printf("Error\n");  
14     return 0;  
15 }  
16
```

```
✓ ↗ 📋  
Hello  
Hello  
Parent  
Child  
Child  
Parent
```

3. How many times “Hello” will be printed?

```
main {  
    fork ( );  
    printf (“Hello”);  
}
```

Hello will be printed 2 times.



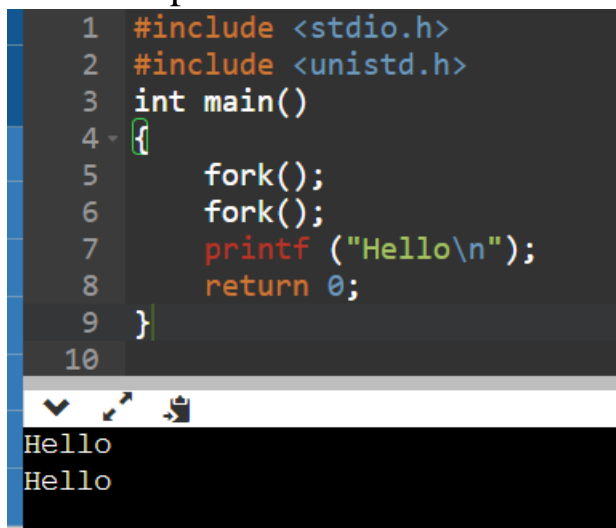
```
1 #include <stdio.h>  
2 #include <unistd.h>  
3 int main()  
4 {  
5     fork();  
6     printf("Hello\n");  
7     return 0;  
8 }  
9
```

Output: Hello  
Hello

4. Count the total number of processes created in the following program-

```
main {  
    fork ( );  
    fork ( );  
    printf (“Hello”);  
}
```

There are  $2^2 - 1 = 3$  child processes and one parent process. So total of 4 processes are there.



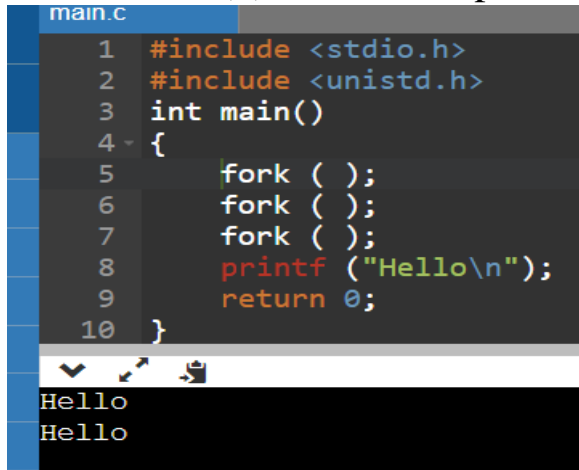
```
1 #include <stdio.h>  
2 #include <unistd.h>  
3 int main()  
4 {  
5     fork();  
6     fork();  
7     printf ("Hello\n");  
8     return 0;  
9 }  
10
```

Output: Hello  
Hello

5. How many child processes will be created?

```
main {  
    fork ( );  
    fork ( );  
    fork ( );  
    printf ("Hello");  
}
```

There are  $2^3 - 1 = 7$  child processes.



```
main.c  
1  #include <stdio.h>  
2  #include <unistd.h>  
3  int main()  
4  {  
5      fork ( );  
6      fork ( );  
7      fork ( );  
8      printf ("Hello\n");  
9      return 0;  
10 }
```

Hello  
Hello

6. How many times “Hello” will be printed?

```
main ( ) {  
    if ( fork ( ) && fork ( ) )  
        fork ( );  
    printf ("Hello");  
}
```

Hello will be printed 4 times.



```
1  #include <stdio.h>  
2  #include <unistd.h>  
3  int main()  
4  {  
5      if ( fork ( ) && fork ( ) ) fork ( );  
6      printf ("Hello\n");  
7      return 0;  
8  }  
9
```

Hello  
Hello

7. How many child processes will be created?

```
main ( ) {  
    if ( fork ( ) || fork ( ) )  
        fork ( );  
    printf ("Hello");  
}
```

5 child processes will be created

A screenshot of a code editor showing a C program. The code includes `<stdio.h>` and `<unistd.h>`, and defines a `main` function. Inside `main`, there is an `if` statement: `if ( fork ( ) || fork ( ) ) fork ( );`. This is followed by `printf ("Hello\n");` and `return 0;`. The code is numbered 1 through 9. Below the code editor, a terminal window shows the output "Hello".

```
1 #include <stdio.h>  
2 #include <unistd.h>  
3 int main()  
4 {  
5     if ( fork ( ) || fork ( ) ) fork ( );  
6     printf ("Hello\n");  
7     return 0;  
8 }  
9  
Hello
```