

## **PRACTICAL 6**

Aim: Write a Shell script to say Good morning /Afternoon/ Evening as you log in to system..

Steps 1. Create a file using a vi editor(or any other editor).

Name script file with extension .sh

```
#!/bin/bash
hour=$( date +"%H" )
echo "Vinayak Patel"
echo "2303031050452"
if [ $hour -ge 6 ] && [ $hour -lt 12 ]
then
    echo "Good Morning"
elif [ $hour -ge 12 ] && [ $hour -lt 18 ]
then
    echo "Good Evening"
elif [ $hour -ge 16 ] && [ $hour -lt 24 ]
then
    echo "Good night"
else
    echo "Midnight"
fi
```

```
"date.sh" 18L, 308B written
[root@localhost ~]# ./date.sh
Vinayak Patel
2303031050452
Good Morning
[root@localhost ~]#
```

## **PRACTICAL 7**

Aim: Write a C program to create a child process:

Example of fork in C

CODE :

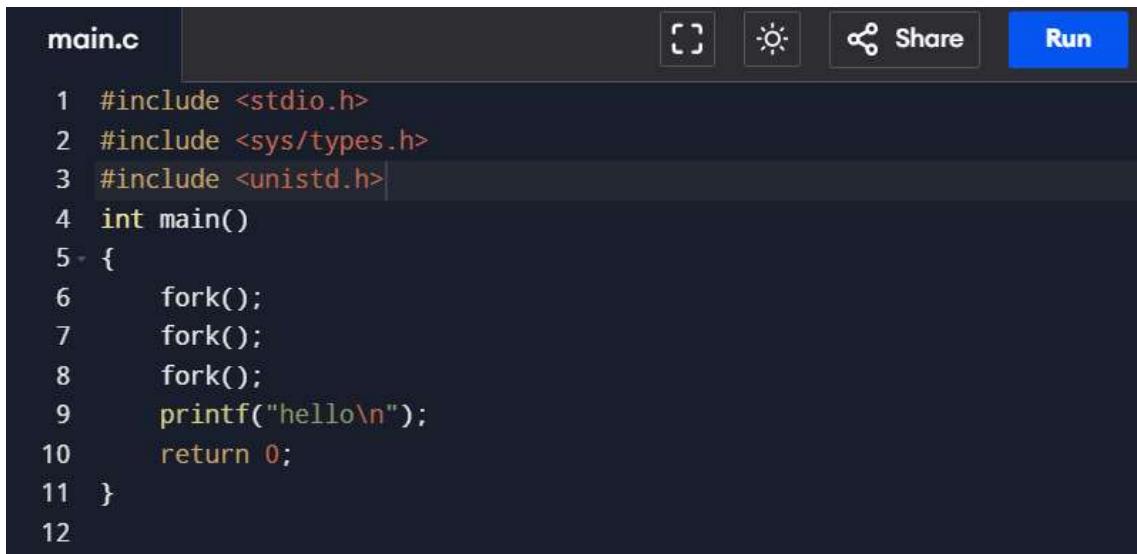
```
main.c | Run | Share | Run
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 #include<stdlib.h>
5 int main()
6 {
7
8     // make two process which run same
9     // program after this instruction
10    pid_t p = fork();
11    if(p<0){
12        perror("fork fail");
13        exit(1);
14    }
15    printf("Hello world!, process_id(pid) = %d \n".getpid());
16    return 0;
17 }
18
```

**Output**

```
Hello world!, process_id(pid) = 806
Hello world!, process_id(pid) = 807

== Code Execution Successful ==
```

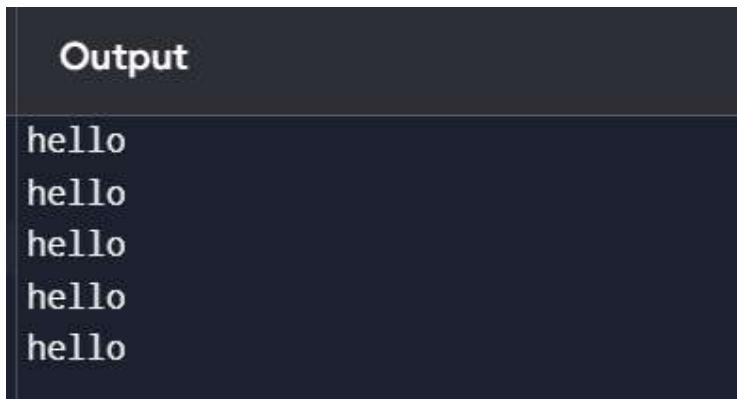
2. CODE:



```
main.c
```

1 #include <stdio.h>  
2 #include <sys/types.h>  
3 #include <unistd.h>  
4 int main()  
5 {  
6 fork();  
7 fork();  
8 fork();  
9 printf("hello\n");  
10 return 0;  
11 }  
12

The screenshot shows a code editor window titled "main.c". The code itself is a C program that includes stdio.h, sys/types.h, and unistd.h. It contains a main function that forks three times and then prints "hello\n" to the standard output. The code editor has a dark theme with syntax highlighting. At the top right, there are several icons: a copy icon, a brightness icon, a share icon, and a blue "Run" button.



### Output

```
hello  
hello  
hello  
hello  
hello
```

The screenshot shows a terminal window titled "Output". It displays five lines of text, each containing the word "hello" followed by a new line character. This output corresponds to the five fork operations performed in the C program, each printing its own "hello" message to the terminal.