

We can use **time** command for this purpose. The time taken is shown in three forms.

real: Total end to end time taken by program/command

user: Time taken in user mode.

sys: Time taken in kernel mode


A Command Example (Time taken by ls-l):

```
$ time ls -l
```

The above command runs "ls -l" and shows contents of current directory followed by the time taken by command "ls -l".

A program example (Time taken by fib(30)):

let us consider below program.



```
#include<stdio.h>
int fib(int n)
{
    if (n <= 1)
        return n;
    return fib(n-1) + fib(n-2);
}

int main ()
{
    printf("Fibonacci Number is %d", fib(30));
    return 0;
}
```

```
// Compiling above program on shell
~$ gcc fib.c
```

```
// Running the generated executable with time
~$ time ./a.out
Fibonacci Number is 832040
real    0m0.017s
user    0m0.017s
sys     0m0.000s
```

Note: 0.017 seconds (shown with real) is total time taken by program.