OPERATING SYSTEM

LAB MANUAL: 6

EXAMPLE:

SOURCE CODE:

```
#include <stdio.h>
#include <sys/stat.h>
/* defines S_IREAD &amp;amp; S_IWRITE */
int main()
{
   int fd;
   fd = creat("datafile.dat", S_IREAD | S_IWRITE);
   if (fd == -1)
   printf("Error in opening datafile.dat\n");
   else
   {
      printf("datafile.dat opened for read/write access\n");
      printf("datafile.dat is currently empty\n");
      }
      close(fd);

}
```

INPUT:

main.c

Run

```c
#include <stdio.h>
#include <sys/stat.h>
/* defines S_IREAD &amp;amp; S_IWRITE */
int main()
{
    int fd;
    fd = creat("datafile.dat", S_IREAD | S_IWRITE);
    if (fd == -1)
    printf("Error in opening datafile.dat\n");
    else
    {
        printf("datafile.dat opened for read/write access\n");
        printf("datafile.dat is currently empty\n");
        }
        close(fd);

}
```

OUTPUT:

Output

```
/tmp/vvalKNWTIf.o
Error in opening datafile.dat
```

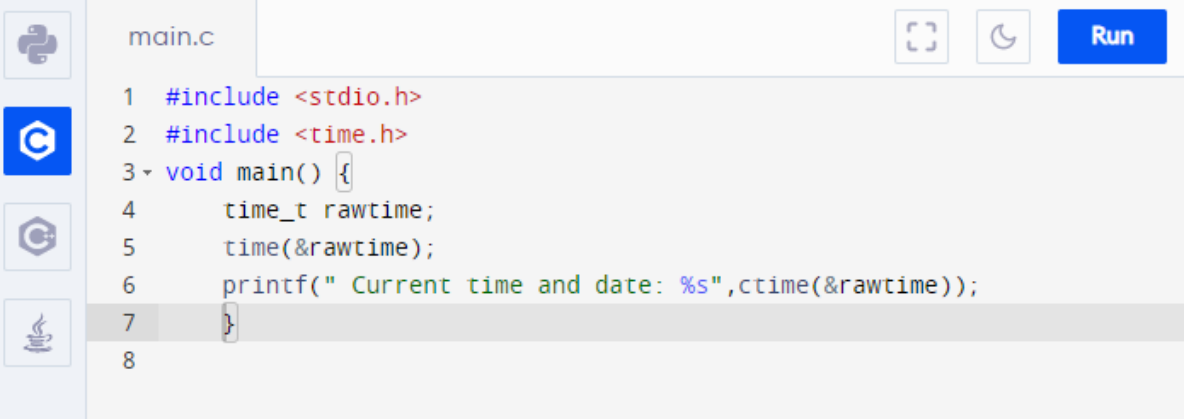
EXAMPLE:

An example is: /* my_date.c print the current date and time in a format similar to the output of the date command.

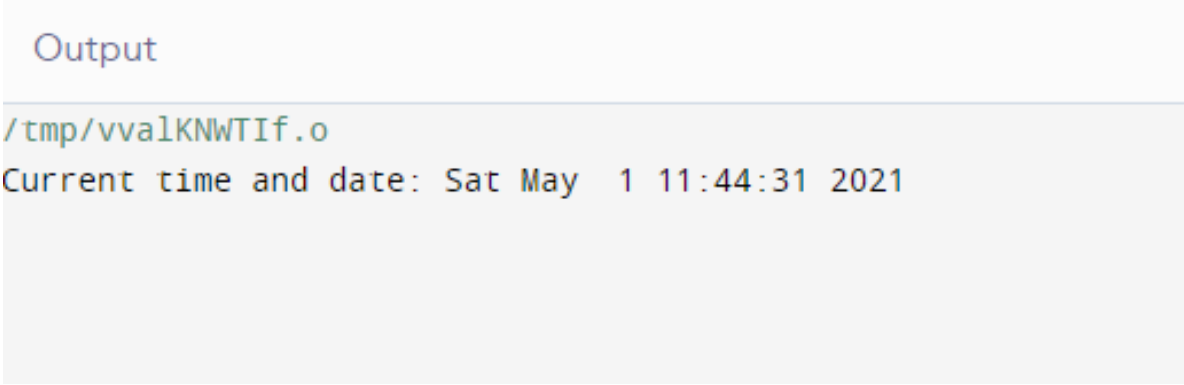
SOURCE CODE:

```
#include <stdio.h>
#include <time.h>
void main() {
   time_t rawtime;
   time(&rawtime);
   printf(" Current time and date: %s",ctime(&rawtime));
   }
```

INPUT:



OUTPUT:



LAB TASK: 1

Compute the Fibonacci series using inter process communication through pipe ().

SOURCE CODE:

```
#include<stdio.h>
main()
{
   int pid;
   int p1[2],p2[2];
   pipe(p1);
   pipe(p2);


   int b,n,i,f1,f2;int ar[30],br[30];
   pid=fork();
   if(pid==0)
   {
      printf("enter count:");
      fflush(stdin);
      scanf("%d",&n);
      close(p1[0]);
      write(p1[1],&n,4);

      close(p2[1]);
      read(p2[0],br,30*sizeof(int));


      printf("\nFibonacci:\n");
      for(i=0;i<n;i++)
         printf("%d\n",br[i]);

   }
```

```
else if(pid>0)
{
close(p1[1]);


read(p1[0],&b,4);


//printf("count is:%d",b);

f1=0,f2=1;


ar[0]=0;


ar[1]=1;
int i;


for(i=2;i<b;i++)


{
   int f3=f1+f2;


   f1=f2;
   f2=f3;


   ar[i]=f3;
}
close(p2[0]);
write(p2[1],ar,30*sizeof(int));


}
```

INPUT:

main.c

Run

```c
#include<stdio.h>
main()
{
    int pid;
    int p1[2],p2[2];
    pipe(p1);
    pipe(p2);
    int b,n,i,f1,f2;int ar[30],br[30];
    pid=fork();
    if(pid==0)
    {
        printf("enter count:");
        fflush(stdin);
        scanf("%d",&n);
        close(p1[0]);
        write(p1[1],&n,4);

        close(p2[1]);
```

main.c

Run

```c
        read(p2[0],br,30*sizeof(int));
        printf("\nFibonacci:\n");
        for(i=0;i<n;i++)
            printf("%d\n",br[i]);

    }
    else if(pid>0)
    {
    close(p1[1]);
    read(p1[0],&b,4);
    //printf("count is:%d",b);

    f1=0,f2=1;
    ar[0]=0;
    ar[1]=1;
    int i;
    for(i=2;i<b;i++)
    {
```

```
37          int f3=f1+f2;
38          f1=f2;
39          f2=f3;
40          ar[i]=f3;
41      }
42      close(p2[0]);
43      write(p2[1],ar,30*sizeof(int));
44      }
45  }
```

OUTPUT:

```
/tmp/vf21U9rIbu.o
enter count:10

Fibonacci:
0
1
1
2
3
5
8
13
21
34
```

----------THE END ----------