



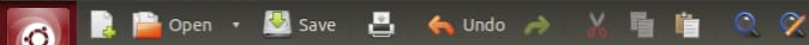
programos.c x

```
#include <stdio.h>
#include<unistd.h>
#include<sys/types.h>
int main()
{
    pid_t p;
    printf("before fork\n");
    p=fork();
    if(p==0)
    {
        printf("I am child having id %d\n",getpid());
        printf("My parent's id is %d\n",getpid());
    }
    else
    {
        printf("M child's id is %d\n",p);
        printf("I am parent having id %d\n",getpid());
    }
    printf("common\n");
}
```

guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF: ~/Desktop

```
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~$ cd Desktop
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ cc programas.c
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out
before fork
M child's id is 3488
I am parent having id 3487
common
I am child having id 3488
My parent's id is 3488
common
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$
```

```
{
printf("M child's id is %d\n",p);
printf("I am parent having id %d\n",getpid());
}
printf("common\n");
}
```



Filecopying.c x copyfile.c x hello.c x

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    FILE *fptr1, *fptr2;
    char filename[100], c;
    printf("Enter the filename to open for reading \n");
    scanf("%s", filename);
    fptr1 = fopen(filename, "r");
    if (fptr1 == NULL){
        printf("Cannot open file %s \n", filename);
        exit(0);
    }
    printf("Enter the filename to open for writing \n");
    scanf("%s", filename);
    fptr2 = fopen(filename, "w");
    if (fptr2 == NULL){
        printf("Cannot open file %s \n", filename);
        exit(0);
    }
    c = fgetc(fptr1);
    while (c != EOF){
        fputc(c, fptr2);
        c = fgetc(fptr1);
    }
    printf("\nContents copied to %s", filename);
    fclose(fptr1);
    fclose(fptr2);
    return 0;
}
```

```
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF: ~/Desktop
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~$ cd Desktop
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ cc copyfile.c
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out
Enter the filename to open for reading
file3.txt
Cannot open file file3.txt
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ cc copyfile.c
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out
Enter the filename to open for reading
filecopying.c
Enter the filename to open for writing
hello.c
Contents copied to hello.cguest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$
```

```
}
    c = fgetc(fp1);
}
printf("\nContents copied to %s", filename);
fclose(fp1);
fclose(fp2);
return 0;
}
```

(globals)

Project C FCFSOS.C

```
1 #include<stdio.h>
2 int main()
3 {
4     int bt[10]={0},at[10]={0},tat[10]={0},wt[10]={0},ct[10]={0};
5     int n,sum=0;
6     float totalTAT=0,totalWT=0;
7     printf("Enter number of processes. ");
8     scanf("%d",&n);
9     printf("Enter arrival time and burst time for each process\n\n");
10    for(int i=0;i<n;i++)
11    {
12        printf("Arrival time of process[%d] ",i+1);
13        scanf("%d",&at[i]);
14        printf("Burst time of process[%d] ",i+1);
15        scanf("%d",&bt[i]);
16        printf("\n");
17    }
18    for(int j=0;j<n;j++)
19    {
20        sum=bt[j];
21        ct[j]=sum;
22    }
23    for(int k=0;k<n;k++)
24    {
25        tat[k]=ct[k]-at[k];
26        totalTAT+=tat[k];
27    }
28    for(int k=0;k<n;k++)
29    {
30        wt[k]=tat[k]-bt[k];
31        totalWT+=wt[k];
32    }
33    printf("Solution: \n\n");
34    printf("P#\t AT\t BT\t CT\t TAT\t WT\t\n\n");
35    for(int i=0;i<n;i++)
36    {
37        printf("P%d\t %d\t %d\t %d\t %d\t %d\n",i+1,at[i],bt[i],ct[i],tat[i],wt[i]);
38    }
39    printf("\n\nAverage Turnaround Time = %f\n",totalTAT/n);
40    printf("Average WT = %f\n",totalWT/n);
41    return 0;
42 }
```

Arrival time of process[1] 2
Burst time of process[1] 25
Arrival time of process[2] 12
Burst time of process[2] 12
Arrival time of process[3] 18
Burst time of process[3] 6
Arrival time of process[4] 21
Burst time of process[4] 18

Solution:

P#	AT	BT	CT	TAT	WT
P1	2	25	25	23	-2
P2	12	12	37	25	13
P3	18	6	43	25	19
P4	21	18	61	40	22

Average Turnaround Time = 28.250000

Average WT = 13.000000

Process exited after 37.97 seconds with return value 0

Press any key to continue . . .

(globals)

Project C FCFSOS.C x SJFOS.C x

```
1 #include <stdio.h>
2 int main()
3 {
4     int A[100][4];
5     int i, j, n, total = 0, index, temp; float avg_wt, avg_tat;
6     printf("Enter number of process: "); scanf("%d", &n);
7     printf("Enter Burst Time:\n");
8     for (i = 0; i < n; i++) {
9         printf("P%d: ", i + 1); scanf("%d", &A[i][1]); A[i][0] = i + 1;
10    }
11    for (i = 0; i < n; i++) {
12        index = i;
13        for (j = i + 1; j < n; j++)
14            if (A[j][1] < A[index][1]) index = j;
15        temp = A[i][1]; A[i][1] = A[index][1]; A[index][1] = temp;
16        temp = A[i][0];
17        A[i][0] = A[index][0]; A[index][0] = temp;
18    }
19    A[0][2] = 0;
20    for (i = 1; i < n; i++) {
21        A[i][2] = 0;
22        for (j = 0; j < i; j++)
23            A[i][2] += A[j][1];
24        total += A[i][2];
25    }
26    avg_wt = (float)total / n; total = 0;
27    printf("P BT WT TAT\n"); for (i = 0; i < n; i++) {
28        A[i][3] = A[i][1] + A[i][2];
29        total += A[i][3];
30        printf("P%d %d %d %d\n", A[i][0], A[i][1], A[i][2], A[i][3]);
31    }
32    avg_tat = (float)total / n;
33    printf("Average Waiting Time= %f", avg_wt); printf("\nAverage Turnaround Time= %f", avg_tat);
34 }
```

Enter number of process: 4

Enter Burst Time:

P1: 25

P2: 12

P3: 6

P4: 18

P BT WT TAT

P3 6 0 6

P2 12 6 18

P4 18 18 36

P1 25 36 61

Average Waiting Time= 15.000000

Average Turnaround Time= 30.250000

Process exited after 18.61 seconds with return value 0

Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Console

Enter number of process:4

Enter Burst Time:

p1:25

p2:12

p3:6

p4:18

Process	Burst Time	Waiting Time	Turnaround Time
p3	6	0	6
p2	12	6	18
p4	18	18	36
p1	25	36	61

Average Waiting Time=15.000000

Average Turnaround Time=30.250000

Process exited after 12.17 seconds with return value 0

Press any key to continue . . . █