

22bps1059

first

```
File Edit View Search Terminal Help
1 #include <sys/wait.h>
2 #include <stdlib.h>
3 #include <stdio.h>
4
5 void pr_exit(int status) {
6     if(WIFEXITED(status)) printf("normal termination : %d\n", WEXITSTATUS(status));
7     else if(WIFSIGNALED(status)) printf("abnormal termination: %d\n", WTERMSIG(status));
8     else if (WIFSTOPPED(status)) printf("child stopped %d\n", WSTOPSIG(status));
9 }
10 pid_t fork();
11 int main() {
12     pid_t pid;
13     int status;
14
15     // process 1 (just exits)
16     if ( ( pid=fork() ) < 0)
17         printf("fork error");
18     else if(pid == 0)
19         exit(0);
20
21     if( wait(&status) != pid )
22         printf("wait error");
23
24     pr_exit(status);
25
26     // process 2 (aborts)
27     if( (pid=fork() ) < 0)
28         printf("fork error");
29     else if (pid == 0)
30         abort();
31
32     if( wait(&status) != pid ) printf("wait error");
33
34     pr_exit(status);
35
36     exit(0);
37 }
38 }
one_processExit.c 1,1 ALL
[0] 0:~$
```

```
22bps1059> ./one_processExit
normal termination : 0
abnormal termination: 6
22bps1059> scrot
22bps1059> |
```

"thecuber-ThinkPad-E14" 12:30 29-Aug-23

second

```
File Edit View Search Terminal Help
1 #include <sys/wait.h>
2 #include <stdlib.h>
3 #include <stdio.h>
4
5 void pr_exit(int status) {
6     if(WIFEXITED(status)) printf("normal termination : %d\n", WEXITSTATUS(status));
7     else if(WIFSIGNALED(status)) printf("abnormal termination: %d\n", WTERMSIG(status));
8     else if (WIFSTOPPED(status)) printf("child stopped %d\n", WSTOPSIG(status));
9 }
10 pid_t fork();
11 int main() {
12     pid_t pid;
13     int status;
14
15     if ( ( pid=fork() ) < 0)
16         printf("fork error");
17     else if(pid == 0)
18         printf("waiting process...\n");
19         wait(0);
20         exit(0);
21
22     if( wait(&status) != pid )
23         printf("wait error");
24
25     pr_exit(status);
26     exit(0);
27 }
two_processwait.c 18,36 ALL
"two_processwait.c" 27L, 674B written
[0] 0:~$
```

```
22bps1059> make two_processwait
cc      two_processwait.c  -o two_processwait
22bps1059> ./two_processwait
waiting process...
22bps1059> |
```

"thecuber-ThinkPad-E14" 12:33 29-Aug-23

third

```
File Edit View Search Terminal Help
1 #include <sys/wait.h>
2 #include <stdlib.h>
3 #include <stdio.h>
4 void pr_exit(int status) {
5     if(WIFEXITED(status)) printf("normal termination : %d\n", WEXITSTATUS(status));
6     else if(WIFSIGNALED(status)) printf("abnormal termination: %d\n", WTERMSIG(status));
7     else if (WIFSTOPPED(status)) printf("child stopped %d\n", WSTOPSIG(status));
8 }
9 pid_t fork(); int execvp(); int sleep(int);
10 int main() {
11     pid_t pid;int status;
12     // process 1
13     if ((pid = fork()) < 0) printf("fork error");
14     else if(pid == 0){
15         char *command = "cat";
16         char *args[] = {command,"file.txt", NULL};
17         execvp(command, args);
18         exit(0);
19     }
20     if(wait(&status) != pid) printf("wait error");
21     pr_exit(status);
22     // process 2
23     if((pid = fork()) < 0) printf("fork error");
24     else if (pid == 0) {
25         abort();
26     }
27     if(wait(&status) != pid) printf("wait error");
28     pr_exit(status);
29     // process 3
30     if((pid = fork()) < 0) printf("fork error");
31     else if (pid == 0){
32         exit(0);
33     }
34     if(wait(&status) != pid) printf("wait error");
35     pr_exit(status);
36     // process 4
37     if((pid = fork()) < 0) printf("fork error");
38     else if (pid == 0){
39         sleep(2);
40         exit(0);
41     }
42     if(wait(&status) != pid) printf("wait error");
43     pr_exit(status);
44     exit(0);
three_entirelifecyle.c 22,5 Top
"three_entirelifecyle.c" 46L, 1318B written
[0] 0:~$
```

```
22bps1059> make three_entirelifecyle
cc      three_entirelifecyle.c  -o three_entirelifecyle
22bps1059> ./three_entirelifecyle
this is some file
with text in it
normal termination : 0
abnormal termination: 6
normal termination : 0
normal termination : 0
22bps1059>
```

"thecuber-ThinkPad-E14" 12:35 29-Aug-23

fourth

```
File Edit View Search Terminal Help
1 #include <sys/wait.h>
2 #include <signal.h>
3 #include <stdlib.h>
4 #include <stdio.h>
5
6 void pr_exit(int status) {
7     if(WIFEXITED(status)) printf("normal termination : %d\n", WEXITSTATUS(status));
8     else if(WIFSIGNALED(status)) printf("abnormal termination: %d\n", WTERMSIG(status));
9     else if (WIFSTOPPED(status)) printf("child stopped %d\n", WSTOPSIG(status));
10 }
11
12 int getpid(); int getppid(); int fork(); int sleep(); int pause(); int kill();
13
14 int main() {
15     int processTimes[] = {7,1,2,1,5};
16     int numofprocesses = sizeof(processTimes) / sizeof(processTimes[0]);
17     pid_t processpids[numofprocesses]; /* W: variable 'processpids' set but not used [-Wunused-variable] */
18
19     //initializing processes
20     for(int i = 0; i<numofprocesses; i++) {
21         pid_t pid = fork();
22         int status;
23         processpids[i] = pid;
24
25         if (pid < 0)
26             perror("fork error");
27
28         else if(pid == 0){
29             printf("\nim process %i my pid is %d and my parent is %d\n",i,getpid(),getppid());
30             fflush(stdout);
31             sleep(processTimes[i]);
32             exit(0);
33         }
34
35         if( wait(&status) != pid )
36             printf("wait error\n");
37
38         pr_exit(status);
39     }
40
41     exit(0);
42     return 0;
43 }
fcfs.c 11,0-1 All
[0] 0:~$
```

```
22bps1059> make fcfs
cc      fcfs.c  -o fcfs
22bps1059> ./fcfs

im process 0 my pid is 235973 and my parent is 235972
normal termination : 0

im process 1 my pid is 235974 and my parent is 235972
normal termination : 0

im process 2 my pid is 235975 and my parent is 235972
normal termination : 0

im process 3 my pid is 235976 and my parent is 235972
normal termination : 0

im process 4 my pid is 235977 and my parent is 235972
normal termination : 0
22bps1059>
```

"thecuber-ThinkPad-E14" 12:36 29-Aug-23

fifth

```
1  ///22bps1059
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <sys/types.h>
5  #include <sys/wait.h>
6  #include <unistd.h>
7
8  void swap( int *x, int *y){
9      int c = *x;
10     *x = *y;
11     *y = c;
12 }
13 int main() {
14     // array of arrival time, bursttime, processnumber
15     int processtable[5][3] = { {8,4,0},{4,10,1},{3,5,2},{31,1,3},{12,3,4} };
16     int numprocesses = 5;
17
18     // sorting accoring to bursttime to get sjfirst
19     for(int i = 0; i< numprocesses; i++) {
20         for(int j = 0; j<numprocesses-1; j++) {
21             if(processtable[j][1] > processtable[j+1][1]) {
22                 swap( &processtable[j][0],&processtable[j+1][0] );
23                 swap( &processtable[j][1],&processtable[j+1][1] );
24                 swap( &processtable[j][2],&processtable[j+1][2] );
25             }
26         }
27     }
28
29     printf("new order of processes (see the third number)\n");
30     for(int i = 0; i< numprocesses; i++) {
31         for(int j = 0; j<3; j++) {
32             printf("%d ",processtable[i][j]);
33         }printf("\n");
34     }
35 }
```

```

34     }
35
36     for(int i = 0; i<numprocesses; i++) {
37         int processdecider = processtable[i][2];
38         pid_t pid = fork(); int status;
39         if (pid == -1) {
40             perror("fork error");
41             exit(1);
42         } else if (pid == 0) {
43             sleep(processtable[i][1]);
44             switch(processdecider){
45                 case 0:
46                     fflush(stdout);
47                     printf("im the first process\n");
48                     break;
49
50                 case 1:
51                     fflush(stdout);
52                     printf("im the second process\n");
53                     break;
54
55                 case 2:
56                     fflush(stdout);
57                     printf("im the Third\n");
58                     break;
59
60                 case 3:
61                     fflush(stdout);
62                     printf("im the fourth\n");
63                     break;
64
65                 case 4:
66                     printf("im the fifth\n");
67                     break;
68             }
69             exit(0);
70         }
71         if( wait(&status) != pid )
72             printf("wait error\n");
73     }
74
75     // Wait for child processes using
76     while(wait(NULL) > 0);
77
78     printf("All child processes have finished in shortest job order.\n");
79     return 0;
80 }
81
82

```

```
File Edit View Search Terminal Help
22bps1059> ./sjf
new order of processes (see the third number)
31 1 3
12 3 4
8 4 0
3 5 2
4 10 1
im the fourth
im the fifth
im the first process
im the Third
im the second process
All child processes have finished in shortest job order.
22bps1059> █

[0] 0: bash* "thecuber-ThinkPad-E14" 12:40 29-Aug-23
```