# PART 1

### 程序编写:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
int main() {
   pid_t pid = fork();
   if (pid < 0) {
       fprintf(stderr, "Fork failed\n");
       return 1;
   }
   else if (pid == 0) { // 子进程
       printf("当前进程为子进程\n");
       printf("当前进程 PID: %d\n", getpid());
       printf("父进程 PID: %d\n", getppid());
       int choice;
       printf("请输入返回值: ");
       scanf("%d", &choice);
       printf("子进程退出\n");
       exit(choice);
   }
   else { // 父进程
       printf("当前进程为父进程\n");
       printf("当前进程 PID: %d\n", getpid());
       printf("子进程 PID: %d\n", pid);
       int status;
       // 等待子进程退出,并获取其返回值
       wait(&status);
       if (WIFEXITED(status)) {
           printf("子进程返回值: %d\n", WEXITSTATUS(status));
       printf("父进程退出\n");
   return 0;
}
```

运行效果:

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e3$ ./a
当前进程为父进程
当前进程 PID:4255
子进程 PID:4256
当前进程 PID:4256
公进程 PID:4255
请输入返回值:5
子进程退出
子进程返回值:5
父进程退出
```

## PART 2

将子进程部分程序代码做如下修改:

```
printf("当前进程为子进程\n");
printf("当前进程 PID: %d\n", getpid());
printf("父进程 PID: %d\n", getppid());
printf("子进程退出\n");
execl("/bin/ls", "ls", NULL);
perror("exec");
exit(EXIT_FAILURE);
```

#### 基于此的输出为:

```
baijy@baijy-virtual-machine:~/Desktop/exercise/e3$ ./a
当前进程为父进程
当前进程 PID:4636
子进程 PID:4637
当前进程为子进程
当前进程 PID:4637
父进程 PID:4636
子进程退出
a try2.cpp try.cpp
子进程返回值:0
```

## PART 3

### 查看所有进程

查看进程的命令是 ps ,增加参数 aux ,可以列出所有进程。

| baijy@bai | jy-virt | ual-ma | achine | e:~/Desl |       |     | ise/e3\$ ps | aux   |                    |
|-----------|---------|--------|--------|----------|-------|-----|-------------|-------|--------------------|
| USER      | PID     | %CPU   | %MEM   | VSZ      | RSS   | TTY | STAT        | START | TIME COMMAND       |
| root      | 1       | 0.2    | 0.2    | 101344   | 11852 | ?   | Ss          | 08:06 | 0:07 /sbin/init au |
| root      | 2       | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [kthreadd]    |
| root      | 3       | 0.0    | 0.0    | 0        | 0     | ?   | I<          | 08:06 | 0:00 [rcu_gp]      |
| root      | 4       | 0.0    | 0.0    | 0        | 0     | ?   | I<          | 08:06 | 0:00 [rcu_par_gp]  |
| root      | 5       | 0.0    | 0.0    | 0        | 0     | ?   | I<          | 08:06 | 0:00 [slub_flushwq |
| root      | 6       | 0.0    | 0.0    | 0        | 0     | ?   | I<          | 08:06 | 0:00 [netns]       |
| root      | 7       | 0.2    | 0.0    | 0        | 0     | ?   | I           | 08:06 | 0:07 [kworker/0:0- |
| root      | 11      | 0.0    | 0.0    | 0        | 0     | ?   | I<          | 08:06 | 0:00 [mm_percpu_wo |
| root      | 12      | 0.0    | 0.0    | 0        | 0     | ?   | I           | 08:06 | 0:00 [rcu_tasks_kt |
| root      | 13      | 0.0    | 0.0    | 0        | 0     | ?   | I           | 08:06 | 0:00 [rcu_tasks_ru |
| root      | 14      | 0.0    | 0.0    | 0        | 0     | ?   | I           | 08:06 | 0:00 [rcu_tasks_tr |
| root      | 15      | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [ksoftirqd/0] |
| root      | 16      | 0.0    | 0.0    | 0        | 0     | ?   | I           | 08:06 | 0:01 [rcu_preempt] |
| root      | 17      | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [migration/0] |
| root      | 18      | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [idle_inject/ |
| root      | 19      | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [cpuhp/0]     |
| root      | 20      | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [cpuhp/1]     |
| root      | 21      | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [idle_inject/ |
| root      | 22      | 0.0    | 0.0    | 0        | 0     | ?   | S           | 08:06 | 0:00 [migration/1] |
|           | 22      |        | 0 0    | ^        | ^     | -   | _           | 00.00 | 0.00 [] [1: 1/4]   |

## 查看进程资源消耗

使用 top 命令可以查看资源消耗

```
top - 08:58:17 up 51 min, 1 user, load average: 0.20, 0.39, 0.36
Tasks: 294 total, 2 running, 292 sleeping, 0 stopped, 0 zombie %Cpu(s): 5.2 us, 1.5 sy, 0.0 ni, 91.7 id, 0.0 wa, 0.0 hi, 1.7 si, 0.0 st
MiB Mem : 3870.6 total,
                           916.0 free, 1506.2 used, 1448.4 buff/cache
                                             0.0 used.
          2140.0 total,
                            2140.0 free,
                                                         2069.9 avail Mem
MiB Swap:
                PR NI VIRT
                                        SHR S %CPU %MEM
                                                                TIME+ COMMAND
    PID USER
                                   RES
                      0 784940 78304 54636 R
   4181 baijy
                 20
                                                               1:14.25 gedit
                                                  6.0
                                                        2.0
                                                5.6
1.0
0.7
                                                       6.9
                                                             1:51.38 gnome-s+
                20 0 4339148 272576
20 0 1783976 397500
                      0 4339148 272576 137816 S
   1264 baijy
                                                             0:38.78 mysqld
                                        35840 S
                                                      10.0
    986 mysql
   563 systemd+ 20 0 14836 6784
3801 baijy 20 0 604468 58676
4568 root 20 0 0 0
                                                       0.2
                                                             0:07.59 systemd+
                                        6016 S
                                                 0.7
                                                             0:11.49 gnome-t+
                                        44188 S
                                                       0.0 0:01.34 kworker+
                                        0 I
                                                 0.7
   4673 baijy
                20 0 16300
                                   4352
                                         3584 R
                                                       0.1 0:00.05 top
                                                 0.7
    245 root
                20 0 0
                                   0
                                          0 I
                                                 0.3 0.0 0:02.17 kworker+
    584 root
                20 0 246732
                                 9216
                                        7936 S 0.3 0.2 0:11.77 vmtoolsd
     1 root
                 20 0 101344 11852
                                         8268 S 0.0 0.3 0:07.34 systemd
                                          0 S
                 20 0
                                                 0.0 0.0
     2 root
                                                             0:00.10 kthreadd
                                           0 I
                 0 -20
     3 root
                              0
                                     0
                                                 0.0 0.0
                                                             0:00.00 rcu_gp
                                                       0.0
     4 root
                  0 -20
                              0
                                     0
                                            0 I
                                                  0.0
                                                              0:00.00 rcu_par+
      5 root
                  0 -20
                              0
                                     0
                                            0 I
                                                  0.0
                                                        0.0
                                                               0:00.00 slub_fl+
                  0 -20
                                            0 I
     6 root
                              0
                                     0
                                                  0.0
                                                        0.0
                                                              0:00.00 netns
                     0
                                                               0:07.91 kworker+
                                            0 I
      7 root
                 20
                              0
                                     0
                                                  0.0
                                                        0.0
                                                       0.0
                 0 -20
                             0
                                     0
                                            0 I
                                                  0.0
                                                             0:00.00 mm_perc+
     11 root
```

# 定位并执行指定操作

可以使用 option \$(pgrep\_process\_id) 的方式,来通过进程名字定位到进程,并进行特定操作。

## 修改优先级

修改优先级可以用命令 renice

# 杀死进程

杀死进程用 kill 操作。

如果无法正常响应,可以加入参数-9来强制杀死。