

4.5 实验 4-5：设置优先级

1. 实验目的

了解和熟悉 Linux 中 `getpriority()` 和 `setpriority()` 系统调用的用法。

2. 实验要求

1) 写一个用户进程，使用 `setpriority()` 来修改进程的优先级，然后使用 `getpriority()` 函数来验证。

2) 可以通过一个 `for` 循环来依次修改进程的优先级（-20~19）。

3. 实验步骤

下面是本实验的实验步骤。

进入本实验的参考代码目录进行交叉编译。

```
cd /home/lab466/runninglinuxkernel_4.0/rlk_lab/rlk_basic/chapter_8/lab5
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabi-
```

编译 `test` 测试 app。然后把它拷贝到 `runninglinuxkernel_4.0/kmodules` 目录下面。

```
arm-linux-gnueabi-gcc process_priority.c -o process_priority --static
cp process_priority /home/lab466/runninglinuxkernel_4.0/kmodules
```

启动 `QEMU+runninglinuxkernel`。最好另外开一个窗口，运行：

```
sudo su
cd /home/lab466/runninglinuxkernel_4.0
sh run.sh arm32
```

进入本实验的参考代码。

```
cd /mnt
```

运行测试程序。

```
./process_priority
```

```
Examining priorities forPID = 1964
Previous Requested Assigned
    0         -20      -20
   -20        -18      -18
   -18        -16      -16
   -16        -14      -14
   -14        -12      -12
   -12        -10      -10
```

-10	-8	-8
-8	-6	-6
-6	-4	-4
-4	-2	-2
-2	0	0
0	2	2
2	4	4
4	6	6
6	8	8
8	10	10
10	12	12
12	14	14
14	16	16
16	18	18

```

benshushu:lab5# ./process_priority
Examining priorities forPID = 1964
Previous Requested Assigned
0 -20 -20
-20 -18 -18
-18 -16 -16
-16 -14 -14
-14 -12 -12
-12 -10 -10
-10 -8 -8
-8 -6 -6
-6 -4 -4
-4 -2 -2
-2 0 0
0 2 2
2 4 4
4 6 6
6 8 8
8 10 10
10 12 12
12 14 14
14 16 16
16 18 18

```

4. 实验代码

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/time.h>
5 #include <sys/resource.h>
6 #include <errno.h>
7
8 int main(int argc, char *argv[])
9 {
10     pid_t mypid;
11     int old_prio, new_prio, i, rc;
12
13     if (argc > 1) {
14         mypid = atoi(argv[1]);
15     } else {
16         mypid = getpid();
17     }
18
19     printf("\nExamining priorities forPID = %d \n", mypid);

```

```
20  printf("%10s%10s%10s\n", "Previous", "Requested", "Assigned");
21
22  for (i = -20; i < 20; i += 2) {
23
24      old_prio = getpriority(PRIO_PROCESS, (int)mypid);
25      rc = setpriority(PRIO_PROCESS, (int)mypid, i);
26      if (rc)
27          fprintf(stderr, "setpriority() failed ");
28
29      /* must clear errno before call to getpriority
30       because -1 is a valid return value */
31      errno = 0;
32
33      new_prio = getpriority(PRIO_PROCESS, (int)mypid);
34      printf("%10d%10d%10d\n", old_prio, i, new_prio);
35
36  }
37  exit(EXIT_SUCCESS);
38}
```