1、signal设置信号处理行为，演示不同信号到来时，后来的信号会打断原有信号的信号处理函数效果

- 在处理信号2的过程中来了信号3，此时则会先执行信号3，等信号3执行完毕再执行信号2

```
^Cbefore sleep the signal is 2
^\before sleep the signal is 3
after sleep the signal is 3
after sleep the signal is 2
```

2、signal设置信号处理行为，演示相同信号到来时，无法打断本信号的效果

- 在处理信号2的过程中来了信号2，不影响当前信号的执行

```
^Cbefore sleep the signal is 2
^C^C^C^C^C^C^C^C^C^Cafter sleep the signal is 2
before sleep the signal is 2
after sleep the signal is 2
```

3、sigaction实现信号不断重入的执行效果(来回打断的效果)

```
^\^C^\^C^\^Cafter sleep the signal is 2
before sleep the signal is 2
after sleep the signal is 2
after sleep the signal is 3
before sleep the signal is 3
after sleep the signal is 3
```

4、sigaction实现2号信号，屏蔽3号信号的执行效果，同时在2号信号的信号处理流程中，判断 3号信号是否被挂起

```
lcrsgk@ubuntu:~/day36/homework04$ ./homework04
^Cthe signal number is 2
^C^C^C
the 2 signal end
the signal number is 2
the 2 signal end
^Cthe signal number is 2
the 2 signal end
^\Quit (core dumped)
lcrsgk@ubuntu:~/day36/homework04$
```

5、通过sigaction，打印发送信号的发送者的pid和uid。

```
gcc      homework05.c   -o homework05
dcrsgk@ubuntu:~/day36/homework05$ ./homework05
^Cthr signal pid = 0,uid = 0
^\
^Cthr signal pid = 0,uid = 0
^Cthr signal pid = 0,uid = 0
^Cthr signal pid = 0,uid = 0
^Cthr signal pid = 0,uid = 0
^Cthr signal pid = 0,uid = 0
^Cthr signal pid = 0,uid = 0
```

6、通过sigprocmask阻塞2号信号，睡眠5秒后，解除阻塞，2号信号得到执行；在睡眠后，解除阻塞之前，通过sigpending检测是否有信号挂起

- sigprocmask没听懂 周末补上

7、使用真实计时器，实用计时器，虚拟计时器，统计程序执行时间，在实用计时器及虚拟计时器设定计时后，先睡眠，再让程序处于while（1）

- 竞争时间可以查看cpu占用率

```
ID USER        PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
24 dcrsgk      20   0    4504    736    676 R  49.7   0.0   0:19.69 while
25 dcrsgk      20   0    4504    796    732 R  49.7   0.0   0:10.42 vir_count
```

- 真实计时器：

```
dcrsgk@ubuntu:~/day36/homework07$ ./time_count
now time is:
now time = Thu Apr 23 22:44:50 2020

now time = Thu Apr 23 22:44:51 2020

now time = Thu Apr 23 22:44:52 2020

now time = Thu Apr 23 22:44:53 2020

now time = Thu Apr 23 22:44:54 2020

now time = Thu Apr 23 22:44:55 2020

now time = Thu Apr 23 22:44:56 2020

now time = Thu Apr 23 22:44:57 2020

^C
```
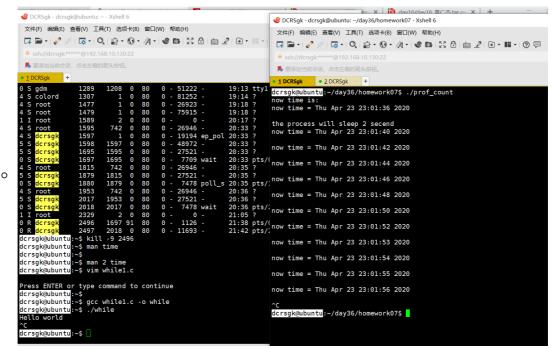
- 实用计时器
  - 在运行时用while1抢时间片造成运行时间变长

- 虚拟计时器

the process will sleep 3 secend
now time = Thu Apr 23 23:09:50 2020

now time = Thu Apr 23 23:09:52 2020

now time = Thu Apr 23 23:09:54 2020

now time = Thu Apr 23 23:09:56 2020

now time = Thu Apr 23 23:09:58 2020

now time = Thu Apr 23 23:10:00 2020

now time = Thu Apr 23 23:10:02 2020

now time = Thu Apr 23 23:10:04 2020

now time = Thu Apr 23 23:10:05 2020

此时结束while1