

# 实验 07 进程与作业管理

班级：数据科学与大数据一班

学号：202026203005

姓名：张华

用户名：s13

## 一、实验目的

1. 练习进程与作业管理

## 二、实验要求

1. 填写实验报告，请将关键命令及其结果进行截图(请确保截图中的文字清晰可见)
2. 导出为 pdf 文件，文件名为用户名-姓名-lab07.pdf，在规定截止时间之前上传作业)
3. 以下步骤中所有 s01 请换成你自己的用户名，01 请换成你自己用户名中的序号。

## 三、实验步骤

(1) 使用你的普通用户身份在本地虚拟控制台 tty2、tty3 上登陆，在 tty2 上运行以下命令：

```
(while true; do echo -n A >> s01.log; sleep 1;done)
```

在 tty3 上运行命令：

```
tail -f s01.log
```

请解释所看到的现象。

虚拟机 tty2 默认是桌面，所以使用 tty3、tty4

```
s13@s13-virtual-machine:~$ tty
/dev/tty3
s13@s13-virtual-machine:~$ (while true;do echo -n A >> s13.log;sleep 1;done)
```

```
s13@s13-virtual-machine:~$
s13@s13-virtual-machine:~$ tty
/dev/tty4
s13@s13-virtual-machine:~$ tail -f s13.log
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
s13@s13-virtual-machine:~$
s13@s13-virtual-machine:~$ tty
/dev/tty4
s13@s13-virtual-machine:~$ tail -f s13.log
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA_
```

Tty2 设置一秒打印一个 A

Tty3 中 A 不断增加

(2) 切换回控制台 tty2，暂停当前进程的执行，并切换回控制台 tty3，查看 tail 命令的输出。





[illegible]

(7) 用 **kill** 命令结束 **tty2** 上的所有三个进程，查看后台进程列表，并切换至 **tty3** 查看 **tail** 命令的输出是否停止。然后用快捷键结束 **tail** 命令的执行。

```

s13@si13-virtual-machine:~$ kill %1
s13@si13-virtual-machine:~$ kill %2
[1]  ♦ ♦ ♦ ( while true; do
    echo -n A >> s13.log; sleep 1;
done )
s13@si13-virtual-machine:~$ kill %3
s13@si13-virtual-machine:~$ jobs
[2]+  ♦ ♦ ♦ ( while true; do
    echo -n B >> s13.log; sleep 1;
done )
[3]-  ♦ ♦ ♦ ( while true; do
    echo -n C >> s13.log; sleep 1;
done )
s13@si13-virtual-machine:~$ jobs
s13@si13-virtual-machine:~$

```

[illegible]

(8) 设置好虚拟机网络，使其能上网，然后以管理员身份安装 `xorg-x11-apps` 软件包。

```
s13@s13-virtual-machine:~$ sudo apt-get install x11-apps
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
x11-apps 已经是最新版 (7.7+8)。
x11-apps 已设置为手动安装。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 19 个软件包未被升级。
```

(9) 从 MobaXterm 后台启动 4 次 `oclock` 程序, 令其 `nice` 值分别为 0, 10, 15, -5, 然后用 `ps` 命令查看它们的 `oclock` 值和优先级, 执行什么操作时需要切换至 `root` 用户?

The screenshot displays a Kali Linux desktop environment. On the left, three 'oclock' application windows are visible, each showing a clock face. On the right, a terminal window shows the following sequence of commands and output:

```

root@s13-virtual-machine: ~
s13@s13-virtual-machine:~$
s13@s13-virtual-machine:~$
s13@s13-virtual-machine:~$
s13@s13-virtual-machine:~$
s13@s13-virtual-machine:~$
s13@s13-virtual-machine:~$ nice -0 oclock &
[1] 39021
s13@s13-virtual-machine:~$ nice -10 oclock &
[2] 39022
s13@s13-virtual-machine:~$ nice -15 oclock &
[3] 39023
s13@s13-virtual-machine:~$ nice --5 oclock &
[4] 39024
s13@s13-virtual-machine:~$ nice: 无法设置优先级: 权限不够
su
密码:
root@s13-virtual-machine:/home/s13# nice --5 oclock &
[1] 39034

```

Red arrows point from the terminal output to the clock windows, indicating that the windows are the processes being monitored. The third clock window (bottom left) is highlighted with a red box, corresponding to the error message in the terminal.

```

root@s13-virtual-machine:~# ps -al
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   1000     1125     1113  0  80   0 - 74153 ep_pol  tty2        00:00:19 Xorg
0 S   1000     1433     1113  0  80   0 - 49878 do_sys  tty2        00:00:00 gnome-session-b
4 S   1000    13872     7825  0  80   0 - 5026 do_sel  tty3        00:00:00 bash
4 S   1000    15107    14991  0  80   0 - 4852 do_sel  tty4        00:00:00 bash
0 S   1000    39022     2969  0  90  10 - 2701 do_sys  pts/0       00:00:00 oclock
0 S   1000    39023     2969  0  95  15 - 2701 do_sys  pts/0       00:00:00 oclock
0 S   1000    39024     2969  0  80   0 - 2701 do_sys  pts/0       00:00:00 oclock
4 S    0     39025     2969  0  80   0 - 5018 do_wai  pts/0       00:00:00 su
4 S    0     39026    39025  0  80   0 - 4659 do_wai  pts/0       00:00:00 bash
4 S    0     39034    39026  0  75  -5 - 2701 do_sys  pts/0       00:00:00 oclock
4 R    0     39054    39026  0  80   0 - 5016 -      pts/0       00:00:00 ps
root@s13-virtual-machine:~#

```

将优先级设为负数时需要 root 用户

(10) 用 renice 命令将 4 个 oclock 进程的 nice 值调整为 5，执行什么操作时需要切换至 root 用户？

```

s13@s13-virtual-machine:~$ renice 5 39022
renice: 设置 39022 的优先级失败(process ID): 权限不够
s13@s13-virtual-machine:~$ renice 5 39023
renice: 设置 39023 的优先级失败(process ID): 权限不够
s13@s13-virtual-machine:~$ renice 5 39024
39024 (process ID) 旧优先级为 0, 新优先级为 5
s13@s13-virtual-machine:~$ renice 5 39034
renice: 设置 39034 的优先级失败(process ID): 不允许的操作
s13@s13-virtual-machine:~$

```

将优先级升高和将负数优先设为整数需要 root

```

s13@s13-virtual-machine:~$ su
密码:
root@s13-virtual-machine:/home/s13# cd
root@s13-virtual-machine:~# renice 5 39022
39022 (process ID) 旧优先级为 10, 新优先级为 5
root@s13-virtual-machine:~# renice 5 39023
39023 (process ID) 旧优先级为 15, 新优先级为 5
root@s13-virtual-machine:~# renice 5 39034
39034 (process ID) 旧优先级为 -5, 新优先级为 5
root@s13-virtual-machine:~#

```

```

root@s13-virtual-machine:~# ps -al
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   1000     1125     1113  0  80   0 - 74128 ep_pol  tty2        00:00:19 Xorg
0 S   1000     1433     1113  0  80   0 - 49878 do_sys  tty2        00:00:00 gnome-session-b
4 S   1000    13872     7825  0  80   0 - 5026 do_sel  tty3        00:00:00 bash
4 S   1000    15107    14991  0  80   0 - 4852 do_sel  tty4        00:00:00 bash
0 S   1000    39022     2969  0  85   5 - 2701 do_sys  pts/0       00:00:00 oclock
0 S   1000    39023     2969  0  85   5 - 2701 do_sys  pts/0       00:00:00 oclock
0 S   1000    39024     2969  0  85   5 - 2701 do_sys  pts/0       00:00:00 oclock
4 S    0     39025     2969  0  80   0 - 5018 do_wai  pts/0       00:00:00 su
4 S    0     39026    39025  0  80   0 - 4659 do_wai  pts/0       00:00:00 bash
4 S    0     39034    39026  0  85   5 - 2701 do_sys  pts/0       00:00:00 oclock
4 S    0     39055    39026  0  80   0 - 4927 do_wai  pts/0       00:00:00 su
4 S   1000    39056    39055  0  80   0 - 4812 do_wai  pts/0       00:00:00 bash
4 S    0     39084    39056  0  80   0 - 5018 do_wai  pts/0       00:00:00 su
4 S    0     39085    39084  0  80   0 - 4619 do_wai  pts/0       00:00:00 bash
4 R    0     39097    39085  0  80   0 - 5016 -      pts/0       00:00:00 ps
root@s13-virtual-machine:~#

```

(11) 用 killall 命令结束所有 oclock 进程，打印进程树，同时显示各进程的 PID。

```

root@s13-virtual-machine:~# killall oclock
root@s13-virtual-machine:~#

```

[illegible]