

# 俞楚凡 ICS 5.12 lab

## 1.1

在linux中，使用 **top** 命令可以实时显示系统的性能状态，进程资源消耗等。

在 **top** 的交互页面中按下 **h** 键，可以弹出一个关于 **top** 命令和它提供的各种快捷方式的详细信息。

```
ainfinity@AInfinity: ~  
Help for Interactive Commands - procps-ng 4.0.4  
Window 1:Def: Cumulative mode Off. System: Delay 3.0 secs; Secure mode Off.  
  
Z,B,E,e Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale  
l,t,m,I,0 Toggle: 'l' load avg; 't' task/cpu; 'm' memory; 'I' Irix; '0' zeros  
1,2,3,4,5 Toggle: '1/2/3' cpu/numa views; '4' cpus abreast; '5' P/E-cores  
f,X Fields: 'f' add/remove/order/sort; 'X' increase fixed-width fields  
  
L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right  
R,H,J,C . Toggle: 'R' Sort; 'H' Threads; 'J' Num justify; 'C' Coordinates  
c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify  
x,y . Toggle highlights: 'x' sort field; 'y' running tasks  
z,b . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')  
u,U,o,O . Filter by: 'u'/'U' effective/any user; 'o'/'O' other criteria  
n,#,^O . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'O' other filter(s)  
V,v,F . Toggle: 'V' forest view; 'v' hide/show children; 'F' keep focused  
  
d,k,r,^R 'd' set delay; 'k' kill; 'r' renice; Ctrl+'R' renice autogroup  
^G,K,N,U View: ctl groups ^G; cmdline ^K; environment ^N; supp groups ^U  
Y,!,^E,P Inspect 'Y'; Combine Cpus '!'; Scale time ^E; View namespaces ^P  
W,q Write config file 'W'; Quit 'q'  
( commands shown with '.' require a visible task display window )  
Press 'h' or '?' for help with Windows,  
Type 'q' or <Esc> to continue |
```

要在 **top** 中只显示特定进程的信息，有两种方法。第一种，在命令行页面中获取到进程（以 **cron** 为例）的 **pid**，然后使用这个 **pid** 运行 **top**：

```
ainfinity@AInfinity:~$ pidof cron  
144  
ainfinity@AInfinity:~$ top -p 144
```

```
ainfinity@Alnfinity: ~  
top - 13:44:05 up 10 min, 1 user, load average: 0.01, 0.02, 0.00  
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 7826.7 total, 7203.0 free, 680.0 used, 146.6 buff/cache  
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7146.6 avail Mem  


| PID | USER | PR | NI | VIRT | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND |
|-----|------|----|----|------|------|------|---|------|------|---------|---------|
| 144 | root | 20 | 0  | 4236 | 2660 | 2424 | S | 0.0  | 0.0  | 0:00.00 | cron    |


```

或者，另一种方式是，在top中进行筛选。按 **o** 键输入筛选条件 **COMMAND=cron**，可以显示某个进程名称对应的进程信息：

```
ainfinity@Alnfinity: ~  
top - 13:45:06 up 11 min, 1 user, load average: 0.00, 0.02, 0.00  
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 7826.7 total, 7209.5 free, 673.4 used, 146.8 buff/cache  
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7153.3 avail Mem  


| PID | USER     | PR | NI | VIRT    | RES   | SHR   | S | %CPU | %MEM | TIME+   | COMMAND         |
|-----|----------|----|----|---------|-------|-------|---|------|------|---------|-----------------|
| 1   | root     | 20 | 0  | 21652   | 13080 | 9684  | S | 0.0  | 0.2  | 0:00.25 | systemd         |
| 2   | root     | 20 | 0  | 2616    | 1440  | 1320  | S | 0.0  | 0.0  | 0:00.00 | init-systemd(Ub |
| 7   | root     | 20 | 0  | 2616    | 132   | 132   | S | 0.0  | 0.0  | 0:00.00 | init            |
| 52  | root     | 19 | -1 | 66816   | 17072 | 15916 | S | 0.0  | 0.2  | 0:00.09 | systemd-journal |
| 94  | root     | 20 | 0  | 23980   | 6052  | 4900  | S | 0.0  | 0.1  | 0:00.07 | systemd-udev    |
| 137 | systemd+ | 20 | 0  | 21452   | 11792 | 9600  | S | 0.0  | 0.1  | 0:00.05 | systemd-resolve |
| 138 | systemd+ | 20 | 0  | 91020   | 6520  | 5672  | S | 0.0  | 0.1  | 0:00.03 | systemd-timesyn |
| 144 | root     | 20 | 0  | 4236    | 2660  | 2424  | S | 0.0  | 0.0  | 0:00.00 | cron            |
| 145 | message+ | 20 | 0  | 9596    | 5036  | 4488  | S | 0.0  | 0.1  | 0:00.02 | dbus-daemon     |
| 156 | root     | 20 | 0  | 17976   | 8276  | 7252  | S | 0.0  | 0.1  | 0:00.03 | systemd-logind  |
| 161 | root     | 20 | 0  | 1756096 | 15756 | 9152  | S | 0.0  | 0.2  | 0:00.06 | wsl-pro-service |
| 167 | root     | 20 | 0  | 3160    | 1200  | 1112  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 173 | root     | 20 | 0  | 3116    | 1104  | 1016  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 183 | syslog   | 20 | 0  | 222508  | 7276  | 4444  | S | 0.0  | 0.1  | 0:00.03 | rsyslogd        |
| 194 | root     | 20 | 0  | 107012  | 22616 | 13212 | S | 0.0  | 0.3  | 0:00.06 | unattended-upgr |
| 285 | root     | 20 | 0  | 2624    | 120   | 0     | S | 0.0  | 0.0  | 0:00.00 | SessionLeader   |
| 286 | root     | 20 | 0  | 2624    | 128   | 0     | S | 0.0  | 0.0  | 0:00.03 | Relay(287)      |
| 287 | ainfini+ | 20 | 0  | 6204    | 5372  | 3584  | S | 0.0  | 0.1  | 0:00.05 | bash            |
| 288 | root     | 20 | 0  | 6692    | 4640  | 3860  | S | 0.0  | 0.1  | 0:00.00 | login           |
| 380 | ainfini+ | 20 | 0  | 20256   | 11412 | 9336  | S | 0.0  | 0.1  | 0:00.04 | systemd         |
| 381 | ainfini+ | 20 | 0  | 21148   | 1728  | 0     | S | 0.0  | 0.0  | 0:00.00 | (sd-pam)        |
| 396 | ainfini+ | 20 | 0  | 6072    | 5076  | 3484  | S | 0.0  | 0.1  | 0:00.01 | bash            |
| 608 | ainfini+ | 20 | 0  | 9288    | 5284  | 3132  | R | 0.0  | 0.1  | 0:00.00 | top             |


```

```
ainfinity@Alnfinity: ~
top - 13:45:19 up 11 min, 1 user, load average: 0.00, 0.01, 0.00
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7826.7 total, 7195.6 free, 687.3 used, 146.8 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7139.4 avail Mem
add filter #1 (ignoring case) as: [!]FLD?VAL COMMAND=cron

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
    1 root        20   0   21652 13080  9684 S   0.0   0.2   0:00.25 systemd
    2 root        20   0   2616  1440  1320 S   0.0   0.0   0:00.00 init-systemd(Ub
    7 root        20   0   2616   132   132 S   0.0   0.0   0:00.00 init
   52 root        19  -1  66816 17072 15916 S   0.0   0.2   0:00.09 systemd-journal
   94 root        20   0  23980  6052  4900 S   0.0   0.1   0:00.07 systemd-udev
  137 systemd+   20   0  21452 11792  9600 S   0.0   0.1   0:00.05 systemd-resolve
  138 systemd+   20   0  91020  6520  5672 S   0.0   0.1   0:00.03 systemd-timesyn
  144 root        20   0   4236  2660  2424 S   0.0   0.0   0:00.00 cron
  145 message+   20   0   9596  5036  4488 S   0.0   0.1   0:00.02 dbus-daemon
  156 root        20   0  17976  8276  7252 S   0.0   0.1   0:00.03 systemd-logind
  161 root        20   0 1756096 15756  9152 S   0.0   0.2   0:00.06 wsl-pro-service
  167 root        20   0   3160  1200  1112 S   0.0   0.0   0:00.00 agetty
  173 root        20   0   3116  1104  1016 S   0.0   0.0   0:00.00 agetty
  183 syslog      20   0  222508  7276  4444 S   0.0   0.1   0:00.03 rsyslogd
  194 root        20   0 107012 22616 13212 S   0.0   0.3   0:00.06 unattended-upgr
  285 root        20   0   2624   120    0 S   0.0   0.0   0:00.00 SessionLeader
  286 root        20   0   2624   128    0 S   0.0   0.0   0:00.03 Relay(287)
  287 ainfini+    20   0   6204  5372  3584 S   0.0   0.1   0:00.05 bash
  288 root        20   0   6692  4640  3860 S   0.0   0.1   0:00.00 login
  380 ainfini+   20   0  20256 11412  9336 S   0.0   0.1   0:00.04 systemd
  381 ainfini+   20   0  21148  1728    0 S   0.0   0.0   0:00.00 (sd-pam)
  396 ainfini+   20   0   6072  5076  3484 S   0.0   0.1   0:00.01 bash
  608 ainfini+   20   0   9288  5284  3132 R   0.0   0.1   0:00.00 top
```

```
ainfinity@Alnfinity: ~
top - 13:45:30 up 11 min, 1 user, load average: 0.00, 0.01, 0.00
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7826.7 total, 7197.8 free, 685.1 used, 146.8 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7141.5 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
  144 root        20   0   4236  2660  2424 S   0.0   0.0   0:00.00 cron
```

在 `top` 运行时按下组合键 `shift + m` 可以让进程按内存排序。

```
ainfinity@Alnfinity: ~  
top - 13:49:35 up 15 min, 1 user, load average: 0.00, 0.00, 0.00  
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 7826.7 total, 7201.3 free, 680.0 used, 149.9 buff/cache  
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7146.7 avail Mem  


| PID | USER     | PR | NI | VIRT    | RES   | SHR   | S | %CPU | %MEM | TIME+   | COMMAND         |
|-----|----------|----|----|---------|-------|-------|---|------|------|---------|-----------------|
| 1   | root     | 20 | 0  | 21652   | 13080 | 9684  | S | 0.0  | 0.2  | 0:00.26 | systemd         |
| 2   | root     | 20 | 0  | 2616    | 1440  | 1320  | S | 0.0  | 0.0  | 0:00.00 | init-systemd(Ub |
| 7   | root     | 20 | 0  | 2616    | 132   | 132   | S | 0.0  | 0.0  | 0:00.00 | init            |
| 52  | root     | 19 | -1 | 66816   | 17084 | 15928 | S | 0.0  | 0.2  | 0:00.10 | systemd-journal |
| 94  | root     | 20 | 0  | 23980   | 6052  | 4900  | S | 0.0  | 0.1  | 0:00.08 | systemd-udev    |
| 137 | systemd+ | 20 | 0  | 21452   | 11792 | 9600  | S | 0.0  | 0.1  | 0:00.05 | systemd-resolve |
| 138 | systemd+ | 20 | 0  | 91020   | 6520  | 5672  | S | 0.0  | 0.1  | 0:00.03 | systemd-timesyn |
| 144 | root     | 20 | 0  | 4236    | 2660  | 2424  | S | 0.0  | 0.0  | 0:00.00 | cron            |
| 145 | message+ | 20 | 0  | 9596    | 5036  | 4488  | S | 0.0  | 0.1  | 0:00.03 | dbus-daemon     |
| 156 | root     | 20 | 0  | 17976   | 8276  | 7252  | S | 0.0  | 0.1  | 0:00.03 | systemd-logind  |
| 161 | root     | 20 | 0  | 1756096 | 15756 | 9152  | S | 0.0  | 0.2  | 0:00.06 | wsl-pro-service |
| 167 | root     | 20 | 0  | 3160    | 1200  | 1112  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 173 | root     | 20 | 0  | 3116    | 1104  | 1016  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 183 | syslog   | 20 | 0  | 222508  | 7276  | 4444  | S | 0.0  | 0.1  | 0:00.04 | rsyslogd        |
| 194 | root     | 20 | 0  | 107012  | 22616 | 13212 | S | 0.0  | 0.3  | 0:00.06 | unattended-upgr |
| 285 | root     | 20 | 0  | 2624    | 120   | 0     | S | 0.0  | 0.0  | 0:00.00 | SessionLeader   |
| 286 | root     | 20 | 0  | 2624    | 128   | 0     | S | 0.0  | 0.0  | 0:00.04 | Relay(287)      |
| 287 | ainfini+ | 20 | 0  | 6204    | 5372  | 3584  | S | 0.0  | 0.1  | 0:00.05 | bash            |
| 288 | root     | 20 | 0  | 6692    | 4640  | 3860  | S | 0.0  | 0.1  | 0:00.00 | login           |
| 380 | ainfini+ | 20 | 0  | 20256   | 11412 | 9336  | S | 0.0  | 0.1  | 0:00.04 | systemd         |
| 381 | ainfini+ | 20 | 0  | 21148   | 1728  | 0     | S | 0.0  | 0.0  | 0:00.00 | (sd-pam)        |
| 396 | ainfini+ | 20 | 0  | 6072    | 5076  | 3484  | S | 0.0  | 0.1  | 0:00.01 | bash            |
| 618 | ainfini+ | 20 | 0  | 9288    | 5336  | 3184  | R | 0.0  | 0.1  | 0:00.00 | top             |


```

```
ainfinity@Alnfinity: ~  
top - 13:49:42 up 15 min, 1 user, load average: 0.00, 0.00, 0.00  
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 7826.7 total, 7194.4 free, 686.9 used, 149.9 buff/cache  
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7139.7 avail Mem  


| PID | USER     | PR | NI | VIRT    | RES   | SHR   | S | %CPU | %MEM | TIME+   | COMMAND         |
|-----|----------|----|----|---------|-------|-------|---|------|------|---------|-----------------|
| 194 | root     | 20 | 0  | 107012  | 22616 | 13212 | S | 0.0  | 0.3  | 0:00.06 | unattended-upgr |
| 52  | root     | 19 | -1 | 66816   | 17084 | 15928 | S | 0.0  | 0.2  | 0:00.10 | systemd-journal |
| 161 | root     | 20 | 0  | 1756096 | 15756 | 9152  | S | 0.0  | 0.2  | 0:00.06 | wsl-pro-service |
| 1   | root     | 20 | 0  | 21652   | 13080 | 9684  | S | 0.0  | 0.2  | 0:00.26 | systemd         |
| 137 | systemd+ | 20 | 0  | 21452   | 11792 | 9600  | S | 0.0  | 0.1  | 0:00.05 | systemd-resolve |
| 380 | ainfini+ | 20 | 0  | 20256   | 11412 | 9336  | S | 0.0  | 0.1  | 0:00.04 | systemd         |
| 156 | root     | 20 | 0  | 17976   | 8276  | 7252  | S | 0.0  | 0.1  | 0:00.03 | systemd-logind  |
| 183 | syslog   | 20 | 0  | 222508  | 7276  | 4444  | S | 0.0  | 0.1  | 0:00.04 | rsyslogd        |
| 138 | systemd+ | 20 | 0  | 91020   | 6520  | 5672  | S | 0.0  | 0.1  | 0:00.04 | systemd-timesyn |
| 94  | root     | 20 | 0  | 23980   | 6052  | 4900  | S | 0.0  | 0.1  | 0:00.08 | systemd-udev    |
| 287 | ainfini+ | 20 | 0  | 6204    | 5372  | 3584  | S | 0.0  | 0.1  | 0:00.05 | bash            |
| 618 | ainfini+ | 20 | 0  | 9288    | 5336  | 3184  | R | 0.0  | 0.1  | 0:00.00 | top             |
| 396 | ainfini+ | 20 | 0  | 6072    | 5076  | 3484  | S | 0.0  | 0.1  | 0:00.01 | bash            |
| 145 | message+ | 20 | 0  | 9596    | 5036  | 4488  | S | 0.0  | 0.1  | 0:00.03 | dbus-daemon     |
| 288 | root     | 20 | 0  | 6692    | 4640  | 3860  | S | 0.0  | 0.1  | 0:00.00 | login           |
| 144 | root     | 20 | 0  | 4236    | 2660  | 2424  | S | 0.0  | 0.0  | 0:00.00 | cron            |
| 381 | ainfini+ | 20 | 0  | 21148   | 1728  | 0     | S | 0.0  | 0.0  | 0:00.00 | (sd-pam)        |
| 2   | root     | 20 | 0  | 2616    | 1440  | 1320  | S | 0.0  | 0.0  | 0:00.00 | init-systemd(Ub |
| 167 | root     | 20 | 0  | 3160    | 1200  | 1112  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 173 | root     | 20 | 0  | 3116    | 1104  | 1016  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 7   | root     | 20 | 0  | 2616    | 132   | 132   | S | 0.0  | 0.0  | 0:00.00 | init            |
| 286 | root     | 20 | 0  | 2624    | 128   | 0     | S | 0.0  | 0.0  | 0:00.04 | Relay(287)      |
| 285 | root     | 20 | 0  | 2624    | 120   | 0     | S | 0.0  | 0.0  | 0:00.00 | SessionLeader   |


```

## 1.2

在linux中，使用 **CTRL + z** 可以暂时挂起正在运行的作业。比如，使用 **vim** 打开一个文件，利用 **ctrl + z** 来暂停文件：

```
ainfinity@AInfinity: ~/ics/attar × + v
This file contains materials for one instance of the attacklab.
Files:
    ctarget
Linux binary with code-injection vulnerability. To be used for phases
1-3 of the assignment.
    rtarget
Linux binary with return-oriented programming vulnerability. To be
used for phases 4-5 of the assignment.
    cookie.txt
Text file containing 4-byte signature required for this lab instance.
    farm.c
Source code for gadget farm present in this instance of rtarget. You
can compile (use flag -Og) and disassemble it to look for gadgets.
    hex2raw
Utility program to generate byte sequences. See documentation in lab
handout.
~
~
~
~
~
~
"README.txt" 28L, 635B 1,1 All
```

```
ainfinity@AInfinity: ~/ics/attar × + v
94 root      20  0  23980  6052  4900 S  0.0  0.1  0:00.08 systemd-udev
287 ainfini+ 20  0  6204  5372  3584 S  0.0  0.1  0:00.05 bash
618 ainfini+ 20  0  9288  5336  3184 R  0.0  0.1  0:00.14 top
396 ainfini+ 20  0  6072  5076  3484 S  0.0  0.1  0:00.01 bash
145 message+ 20  0  9596  5036  4488 S  0.0  0.1  0:00.03 dbus-daemon
288 root      20  0  6692  4640  3860 S  0.0  0.1  0:00.00 login
144 root      20  0  4236  2660  2424 S  0.0  0.0  0:00.00 cron
381 ainfini+ 20  0  21148  1728   0 S  0.0  0.0  0:00.00 (sd-pam)
2 root       20  0  2616  1440  1320 S  0.0  0.0  0:00.00 init-systemd(Ub
167 root     20  0  3160  1200  1112 S  0.0  0.0  0:00.00 agetty
173 root     20  0  3116  1104  1016 S  0.0  0.0  0:00.00 agetty
7 root       20  0  2616  132   132 S  0.0  0.0  0:00.00 init
286 root     20  0  2624  128   0 S  0.0  0.0  0:00.05 Relay(287)
285 root     20  0  2624  120   0 S  0.0  0.0  0:00.00 SessionLeader

ainfinity@AInfinity:~$ ls
codes ics
ainfinity@AInfinity:~$ cd ics
ainfinity@AInfinity:~/ics$ ls
attacklab bomblab datalab1 lab5 lab6 lab8 lab9
ainfinity@AInfinity:~/ics$ cd attacklab
ainfinity@AInfinity:~/ics/attacklab$ ls
target176 target176.tar
ainfinity@AInfinity:~/ics/attacklab$ cd target176/
ainfinity@AInfinity:~/ics/attacklab/target176$ ls
README.txt attack.o cookie.txt example.d exploit.txt farm.d farm.s rtarget test.c
attack.d attack.s ctarget exploit-raw.txt farm.c farm.o hex2raw rtarget.d test.s
ainfinity@AInfinity:~/ics/attacklab/target176$ vim README.txt

[1]+  Stopped                  vim README.txt
ainfinity@AInfinity:~/ics/attacklab/target176$ |
```

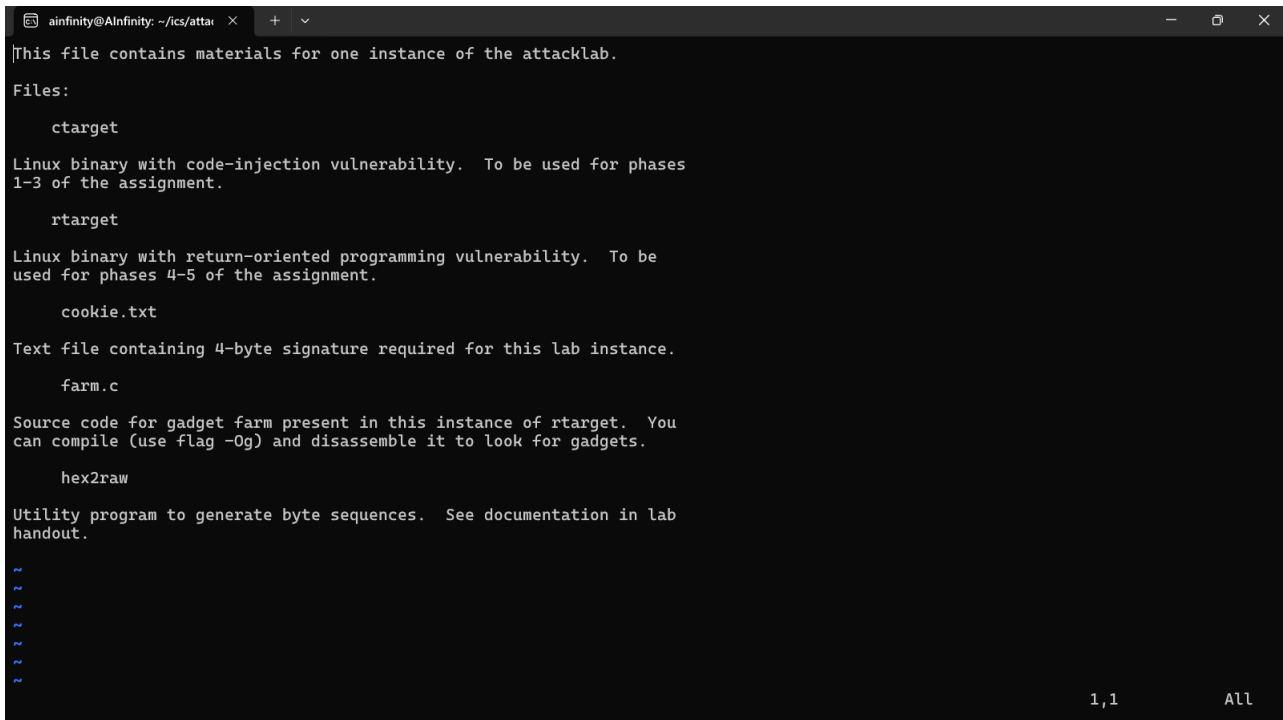
当我按下 `ctrl + z` 后，将会退回到主界面；

此时使用 `jobs` 命令可以列出当前暂停的作业：

```
ainfinity@AInfinity:~/ics/attacklab/target176$ jobs
[1]+  Stopped                  vim README.txt
ainfinity@AInfinity:~/ics/attacklab/target176$ |
```

使用 `fg` 可以恢复暂停的作业。如果只想恢复最近暂停的作业，键入 `fg` 即可。如果想恢复指定作业，可以使用 `fg % + id` 的形式，如 `fg %1`。

```
ainfinity@AInfinity:~/ics/attacklab/target176$ jobs
[1]+  Stopped                  vim README.txt
ainfinity@AInfinity:~/ics/attacklab/target176$ fg %1
```



```
ainfinity@AInfinity: ~/ics/attar × + ∨
This file contains materials for one instance of the attacklab.
Files:
    ctarget
Linux binary with code-injection vulnerability. To be used for phases
1-3 of the assignment.
    rtarget
Linux binary with return-oriented programming vulnerability. To be
used for phases 4-5 of the assignment.
    cookie.txt
Text file containing 4-byte signature required for this lab instance.
    farm.c
Source code for gadget farm present in this instance of rtarget. You
can compile (use flag -Og) and disassemble it to look for gadgets.
    hex2raw
Utility program to generate byte sequences. See documentation in lab
handout.
~
~
~
~
~
~
1,1 All
```

## 1.3

要完成这个任务，首先需要两个终端。

在第一个终端中，运行一个前台任务，例如 `top`。

```
ainfinity@AINfinity: ~  
top - 14:07:48 up 33 min, 1 user, load average: 0.00, 0.00, 0.00  
Tasks: 26 total, 1 running, 25 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 7826.7 total, 7186.7 free, 687.9 used, 163.5 buff/cache  
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7138.8 avail Mem  


| PID | USER     | PR | NI | VIRT    | RES   | SHR   | S | %CPU | %MEM | TIME+   | COMMAND         |
|-----|----------|----|----|---------|-------|-------|---|------|------|---------|-----------------|
| 1   | root     | 20 | 0  | 21652   | 13080 | 9684  | S | 0.0  | 0.2  | 0:00.27 | systemd         |
| 2   | root     | 20 | 0  | 2616    | 1440  | 1320  | S | 0.0  | 0.0  | 0:00.00 | init-systemd(Ub |
| 7   | root     | 20 | 0  | 2616    | 132   | 132   | S | 0.0  | 0.0  | 0:00.00 | init            |
| 52  | root     | 19 | -1 | 66816   | 17124 | 15956 | S | 0.0  | 0.2  | 0:00.12 | systemd-journal |
| 94  | root     | 20 | 0  | 23980   | 6052  | 4900  | S | 0.0  | 0.1  | 0:00.09 | systemd-udev    |
| 137 | systemd+ | 20 | 0  | 21452   | 11792 | 9600  | S | 0.0  | 0.1  | 0:00.05 | systemd-resolve |
| 138 | systemd+ | 20 | 0  | 91020   | 6520  | 5672  | S | 0.0  | 0.1  | 0:00.04 | systemd-timesyn |
| 144 | root     | 20 | 0  | 4236    | 2660  | 2424  | S | 0.0  | 0.0  | 0:00.00 | cron            |
| 145 | message+ | 20 | 0  | 9596    | 5036  | 4488  | S | 0.0  | 0.1  | 0:00.03 | dbus-daemon     |
| 156 | root     | 20 | 0  | 17976   | 8276  | 7252  | S | 0.0  | 0.1  | 0:00.04 | systemd-logind  |
| 161 | root     | 20 | 0  | 1756096 | 15756 | 9152  | S | 0.0  | 0.2  | 0:00.08 | wsl-pro-service |
| 167 | root     | 20 | 0  | 3160    | 1200  | 1112  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 173 | root     | 20 | 0  | 3116    | 1104  | 1016  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 183 | syslog   | 20 | 0  | 222508  | 7276  | 4444  | S | 0.0  | 0.1  | 0:00.04 | rsyslogd        |
| 194 | root     | 20 | 0  | 107012  | 22616 | 13212 | S | 0.0  | 0.3  | 0:00.06 | unattended-upgr |
| 288 | root     | 20 | 0  | 6692    | 4640  | 3860  | S | 0.0  | 0.1  | 0:00.00 | login           |
| 380 | ainfini+ | 20 | 0  | 20256   | 11412 | 9336  | S | 0.0  | 0.1  | 0:00.04 | systemd         |
| 381 | ainfini+ | 20 | 0  | 21148   | 1728  | 0     | S | 0.0  | 0.0  | 0:00.00 | (sd-pam)        |
| 396 | ainfini+ | 20 | 0  | 6072    | 5076  | 3484  | S | 0.0  | 0.1  | 0:00.01 | bash            |
| 640 | root     | 20 | 0  | 2624    | 124   | 0     | S | 0.0  | 0.0  | 0:00.00 | SessionLeader   |
| 641 | root     | 20 | 0  | 2624    | 132   | 0     | S | 0.0  | 0.0  | 0:00.00 | Relay(643)      |
| 643 | ainfini+ | 20 | 0  | 6204    | 5300  | 3580  | S | 0.0  | 0.1  | 0:00.02 | bash            |
| 666 | root     | 20 | 0  | 2624    | 124   | 0     | S | 0.0  | 0.0  | 0:00.00 | SessionLeader   |
| 667 | root     | 20 | 0  | 2624    | 132   | 0     | S | 0.0  | 0.0  | 0:00.00 | Relay(669)      |
| 669 | ainfini+ | 20 | 0  | 6204    | 5216  | 3496  | S | 0.0  | 0.1  | 0:00.01 | bash            |
| 693 | ainfini+ | 20 | 0  | 9288    | 5288  | 3140  | R | 0.0  | 0.1  | 0:00.00 | top             |


```

在第二个终端中，我们将使用 `kill` 命令发送信号来控制这些任务。

首先在第二个终端中查看它的pid。

```
ainfinity@AINfinity: ~  
wsl: 检测到 localhost 代理配置，但未镜像到 WSL。NAT 模式下的 WSL 不支持 localhost 代理。  
ainfinity@AINfinity:~$ pidof top  
693  
ainfinity@AINfinity:~$ |
```

第二步，使用 `kill` 发送 `SIGTERM` 信号。

```
ainfinity@AlInfinity: ~  
wsl: 检测到 localhost 代理配置, 但未镜像到 WSL. NAT 模式下的 WSL 不支持 localhost 代理。  
ainfinity@AlInfinity:~$ pidof top  
693  
ainfinity@AlInfinity:~$ kill -SIGTERM 693  
ainfinity@AlInfinity:~$
```

此时可以看见 **top** 已经退出。

```
ainfinity@AlInfinity: ~  
Tasks: 26 total, 1 running, 25 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 7826.7 total, 7173.9 free, 700.6 used, 163.6 buff/cache  
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7126.1 avail Mem  


| PID | USER     | PR | NI | VIRT    | RES   | SHR   | S | %CPU | %MEM | TIME+   | COMMAND         |
|-----|----------|----|----|---------|-------|-------|---|------|------|---------|-----------------|
| 1   | root     | 20 | 0  | 21652   | 13080 | 9684  | S | 0.0  | 0.2  | 0:00.27 | systemd         |
| 2   | root     | 20 | 0  | 2616    | 1440  | 1320  | S | 0.0  | 0.0  | 0:00.00 | init-systemd(Ub |
| 7   | root     | 20 | 0  | 2616    | 132   | 132   | S | 0.0  | 0.0  | 0:00.00 | init            |
| 52  | root     | 19 | -1 | 66816   | 17128 | 15960 | S | 0.0  | 0.2  | 0:00.12 | systemd-journal |
| 94  | root     | 20 | 0  | 23980   | 6052  | 4900  | S | 0.0  | 0.1  | 0:00.11 | systemd-udev    |
| 137 | systemd+ | 20 | 0  | 21452   | 11792 | 9600  | S | 0.0  | 0.1  | 0:00.05 | systemd-resolve |
| 138 | systemd+ | 20 | 0  | 91020   | 6520  | 5672  | S | 0.0  | 0.1  | 0:00.04 | systemd-timesyn |
| 144 | root     | 20 | 0  | 4236    | 2660  | 2424  | S | 0.0  | 0.0  | 0:00.00 | cron            |
| 145 | message+ | 20 | 0  | 9596    | 5036  | 4488  | S | 0.0  | 0.1  | 0:00.03 | dbus-daemon     |
| 156 | root     | 20 | 0  | 17976   | 8276  | 7252  | S | 0.0  | 0.1  | 0:00.04 | systemd-logind  |
| 161 | root     | 20 | 0  | 1756096 | 15756 | 9152  | S | 0.0  | 0.2  | 0:00.09 | wsl-pro-service |
| 167 | root     | 20 | 0  | 3160    | 1200  | 1112  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 173 | root     | 20 | 0  | 3116    | 1104  | 1016  | S | 0.0  | 0.0  | 0:00.00 | agetty          |
| 183 | syslog   | 20 | 0  | 222508  | 7276  | 4444  | S | 0.0  | 0.1  | 0:00.04 | rsyslogd        |
| 194 | root     | 20 | 0  | 107012  | 22616 | 13212 | S | 0.0  | 0.3  | 0:00.06 | unattended-upgr |
| 288 | root     | 20 | 0  | 6692    | 4640  | 3860  | S | 0.0  | 0.1  | 0:00.00 | login           |
| 380 | ainfini+ | 20 | 0  | 20256   | 11412 | 9336  | S | 0.0  | 0.1  | 0:00.04 | systemd         |
| 381 | ainfini+ | 20 | 0  | 21148   | 1728  | 0     | S | 0.0  | 0.0  | 0:00.00 | (sd-pam)        |
| 396 | ainfini+ | 20 | 0  | 6072    | 5076  | 3484  | S | 0.0  | 0.1  | 0:00.01 | bash            |
| 640 | root     | 20 | 0  | 2624    | 124   | 0     | S | 0.0  | 0.0  | 0:00.00 | SessionLeader   |
| 641 | root     | 20 | 0  | 2624    | 132   | 0     | S | 0.0  | 0.0  | 0:00.00 | Relay(643)      |
| 643 | ainfini+ | 20 | 0  | 6204    | 5300  | 3580  | S | 0.0  | 0.1  | 0:00.02 | bash            |
| 666 | root     | 20 | 0  | 2624    | 124   | 0     | S | 0.0  | 0.0  | 0:00.00 | SessionLeader   |
| 667 | root     | 20 | 0  | 2624    | 132   | 0     | S | 0.0  | 0.0  | 0:00.00 | Relay(669)      |
| 669 | ainfini+ | 20 | 0  | 6204    | 5276  | 3496  | S | 0.0  | 0.1  | 0:00.01 | bash            |
| 693 | ainfini+ | 20 | 0  | 9288    | 5288  | 3140  | R | 0.0  | 0.1  | 0:00.03 | top             |

  
ainfinity@AlInfinity:~$
```

要杀死一个进程组，我们首先要在一个前台中启动一个进程组。我们使用：

```
(sleep 1000 & sleep 1000 &)
```

这将创建一个新的子shell，并在这个shell中后台运行两个sleep命令。随后，我们在可以使用ps命令结合grep筛选出sleep对应的父进程。



```
ainfinity@AlInfinity: ~  
wsl: 检测到 localhost 代理配置, 但未镜像到 WSL. NAT 模式下的 WSL 不支持 localhost 代理。  
ainfinity@AlInfinity:~$ ps  
  PID TTY          TIME CMD  
 1139 pts/0    00:00:00 bash  
 1155 pts/0    00:00:00 ps  
ainfinity@AlInfinity:~$ (sleep 1000 & sleep 1000 &)  
ainfinity@AlInfinity:~$ ps  
  PID TTY          TIME CMD  
 1139 pts/0    00:00:00 bash  
 1161 pts/0    00:00:00 sleep  
 1162 pts/0    00:00:00 sleep  
 1163 pts/0    00:00:00 ps  
ainfinity@AlInfinity:~$ ps -ef | grep 'sleep\\|bash'  
ainfiniti+ 396      288    0 13:33 pts/1    00:00:00 -bash  
ainfiniti+ 1117     641    0 14:14 ?        00:00:00 sleep 1000  
ainfiniti+ 1124     641    0 14:17 ?        00:00:00 sleep 1000  
ainfiniti+ 1125     641    0 14:17 ?        00:00:00 sleep 1000  
ainfiniti+ 1139     1138   0 14:21 pts/0    00:00:00 -bash  
ainfiniti+ 1161     1138   0 14:22 pts/0    00:00:00 sleep 1000  
ainfiniti+ 1162     1138   0 14:22 pts/0    00:00:00 sleep 1000  
ainfiniti+ 1165     1139   0 14:23 pts/0    00:00:00 grep --color=auto sleep\\|bash  
ainfinity@AlInfinity:~$ ps -p 1138  
  PID TTY          TIME CMD  
 1138 ?        00:00:00 Relay(1139)  
ainfinity@AlInfinity:~$ ps -p 1139  
  PID TTY          TIME CMD  
 1139 pts/0    00:00:00 bash  
ainfinity@AlInfinity:~$ |
```

在这张图中，我们可以分析其结构。可以看见，根据最新的时间，我们创建了两个pid分别为1161和1162，pgid为1138的子进程，1138直接关联1139，而1139是一个子**bash**。故而，我们杀死这个进程组的父进程1139，即可杀死整个进程组。

```
ainfinity@AlInfinity: ~  
wsl: 检测到 localhost 代理配置, 但未镜像到 WSL. NAT 模式下的 WSL 不支持 localhost 代理。  
ainfinity@AlInfinity:~$ ps  
  PID TTY          TIME CMD  
 1139 pts/0    00:00:00 bash  
 1155 pts/0    00:00:00 ps  
ainfinity@AlInfinity:~$ (sleep 1000 & sleep 1000 &)  
ainfinity@AlInfinity:~$ ps  
  PID TTY          TIME CMD  
 1139 pts/0    00:00:00 bash  
 1161 pts/0    00:00:00 sleep  
 1162 pts/0    00:00:00 sleep  
 1163 pts/0    00:00:00 ps  
ainfinity@AlInfinity:~$ ps -ef | grep 'sleep\\|bash'  
ainfiniti+ 396      288    0 13:33 pts/1    00:00:00 -bash  
ainfiniti+ 1117     641    0 14:14 ?        00:00:00 sleep 1000  
ainfiniti+ 1124     641    0 14:17 ?        00:00:00 sleep 1000  
ainfiniti+ 1125     641    0 14:17 ?        00:00:00 sleep 1000  
ainfiniti+ 1139     1138   0 14:21 pts/0    00:00:00 -bash  
ainfiniti+ 1161     1138   0 14:22 pts/0    00:00:00 sleep 1000  
ainfiniti+ 1162     1138   0 14:22 pts/0    00:00:00 sleep 1000  
ainfiniti+ 1165     1139   0 14:23 pts/0    00:00:00 grep --color=auto sleep\\|bash  
ainfinity@AlInfinity:~$ ps -p 1138  
  PID TTY          TIME CMD  
 1138 ?        00:00:00 Relay(1139)  
ainfinity@AlInfinity:~$ ps -p 1139  
  PID TTY          TIME CMD  
 1139 pts/0    00:00:00 bash  
ainfinity@AlInfinity:~$ kill -SIGTERM 1139
```

输入kill -SIGKILL 1139后，终端被关闭，计时任务也随之停止。

## 2.1

程序如下：

```
ainfinity@AInfinity: ~/ics/1 × + v
void printID(const char *message) {
    printf("%s: PID = %d, PGID = %d\n", message, getpid(), getpgid(getpid()));
}

int main() {
    // print main pid and pgid
    printID("main process");

    pid_t pid;

    pid = fork();
    if (pid == -1) {
        perror("fork failed");
        return -1;
    }
    else if (pid == 0) {
        printID("child process");
        return 0;
    }

    pid = fork();
    if (pid == -1) {
        perror("fork failed");
        return -1;
    }
    else if (pid == 0) {
        setpgid(0, 0); // set self pgid to self pid
        printID("self pgrouned child process");
        return 0;
    }

    return 0;
}
```

38,0-1 Bot

首先定义了 `printID(const char *message)` 函数用来打印自身的 `pid` 和 `pgid` 信息。随后，首先打印了主进程的信息，随后创建了一个子进程，打印了子进程的信息；最后创建了第二个子进程，并且将其 `pgid` 设置为自身的 `pid`。这里的一个点是，如果 `setpgid` 的两个参数为 0，则默认指向进程自身的 `pid`。最后输出子进程的信息。输出如下：

```
ainfinity@AInfinity:~/ics/lab10$ gcc pg.c -o pg
ainfinity@AInfinity:~/ics/lab10$ ./pg
main process: PID = 1281, PGID = 1281
child process: PID = 1282, PGID = 1281
self pgrouned child process: PID = 1283, PGID = 1283
```

## 2.2

程序如下：

```
int ccount;

void childHandler(int sig) {
    pid_t pid;
    while ((pid = wait(NULL)) > 0) {
        ccount--;
    }
}
```

```
ainfinity@AInfinity: ~/ics/lab1l × + v
}

int main() {
    char input[19];
    fgets(input, sizeof(input), stdin);
    input[strcspn(input, "\n")] = 0; // remove \n

    signal(SIGCHLD, childHandler);

    while (1) {
        ccount = rand() % 3 + 1;

        for (int i = 1; i <= ccount; ++i) {
            pid_t pid = fork();
            if (pid == -1) {
                perror("fork failed");
                return EXIT_FAILURE;
            }
            else if (pid == 0) {
                sleep(1);
                printf("%s\n", input);
                exit(EXIT_SUCCESS);
            }
        }

        sleep(1);
        printf("Processing...\n");

        while (ccount);

        printf("Process finished.\n\n");
    }
    return 0;
}

"process.c" 49L, 771B                                     45,34-48      Bot
```

我们首先获得一个输入字符串，随后进入死循环，每次随机生成1-3个子进程，与父进程同步打印字符串，最后利用childHandler来接受信号，回收子进程。等待子进程全部被回收完毕后，打印Process finished信息，标志回收已经完成。

程序输出如下：

```
ainfinity@AInfinity: ~/ics/lab1l × + v
^C
ainfinity@AInfinity:~/ics/lab1l$ vim process.c
ainfinity@AInfinity:~/ics/lab1l$ ./process
hello process!
hello process!
Processing...
hello process!
Process finished.

hello process!
hello process!
Processing...
Process finished.

hello process!
Processing...
Process finished.

hello process!
hello process!
Processing...
Process finished.

hello process!
hello process!
hello process!
Processing...
Process finished.

hello process!
hello process!
Processing...
Process finished.

^C
ainfinity@AInfinity:~/ics/lab1l$ |
```

## 2.3

改造了程序2.2，现在它具备下列功能：

能够接受控制台输入和kill发送的信号，kill发送SIGUSR1信号后，使用longjmp倒回到输入字符串之前，重新进行输入。发送SIGUSR2信号后，将flag置0，从而跳出while循环，打印退出信息并结束程序。程序如下：

```
    signal(SIGUSR2, usr2Handler);

    setjmp(b);

    fgets(input, sizeof(input), stdin);
    input[strcspn(input, "\n")] = 0; // remove \n

    while (flag) {
        ccount = rand() % 3 + 1;

        for (int i = 1; i <= ccount; ++i) {
            pid_t pid = fork();
            if (pid == -1) {
                perror("fork failed");
                return EXIT_FAILURE;
            }
            else if (pid == 0) {
                sleep(1);
                printf("%s\n", input);
                exit(EXIT_SUCCESS);
            }
        }

        sleep(1);
        printf("Processing...\n");

        while (ccount);

        printf("Process finished.\n\n");
    }

    printf("Exiting!\n");
    return 0;
}
```

我们编译并运行程序：

```
ainfinity@AInfinity:~/ics/lab10$ ./process
hello world!
hello world!
Processing...
hello world!
Process finished.

hello world!
Processing...
hello world!
Process finished.

Processing...
hello world!
Process finished.

|
```

可以看到，程序现在正常运行。

我们打开另外一个终端，获取process进程组的父进程的pid，并向其发送SIGUSR1信号：

```
ainfinity@AInfinity:~$ ps -ef | grep process
ainfini+   1866    1175  0 16:01 pts/3    00:00:00 ./process
ainfini+   1874    1866  0 16:01 pts/3    00:00:00 ./process
ainfini+   1875    1866  0 16:01 pts/3    00:00:00 ./process
ainfini+   1876    1866  0 16:01 pts/3    00:00:00 ./process
ainfini+   1878    1447  0 16:01 pts/0    00:00:00 grep --color=auto process
ainfinity@AInfinity:~$ kill -SIGUSR1 1866
ainfinity@AInfinity:~$ |
```

我们此时可以重新输入字符串，程序将会处理新输入的字符串。

```
ainfinity@AInfinity: ~/ics/lab10 X + v
hello world!
Processing...
hello world!
hello world!
Process finished.

hello world!
hello world!
Processing...
Process finished.

hello world!
hello world!
hello world!
Processing...
Process finished.

hello world!
hello world!
broken by SIGUSR1
Processing...
broken by SIGUSR1
broken by SIGUSR1
broken by SIGUSR1
Process finished.

Processing...
broken by SIGUSR1
broken by SIGUSR1
Process finished.

Processing...
broken by SIGUSR1
Process finished.
```

```
ainfinity@AInfinity:~$ ps -ef | grep process
ainfini+ 1866 1175 0 16:01 pts/3 00:00:00 ./process
ainfini+ 1874 1866 0 16:01 pts/3 00:00:00 ./process
ainfini+ 1875 1866 0 16:01 pts/3 00:00:00 ./process
ainfini+ 1876 1866 0 16:01 pts/3 00:00:00 ./process
ainfini+ 1878 1447 0 16:01 pts/0 00:00:00 grep --color=auto process
ainfinity@AInfinity:~$ kill -SIGUSR1 1866
ainfinity@AInfinity:~$ kill -SIGUSR2 1866
ainfinity@AInfinity:~$ |
```

发送SIGUSR2信号，退出while循环，打印退出信息并退出。

```
ainfinity@AInfinity: ~/ics/lab10 X + v
broken by SIGUSR1
broken by SIGUSR1
broken by SIGUSR1
Processing...
Process finished.

Processing...
broken by SIGUSR1
Process finished.

Processing...
broken by SIGUSR1
Process finished.

Processing...
broken by SIGUSR1
Process finished.

broken by SIGUSR1
broken by SIGUSR1
Processing...
broken by SIGUSR1
Process finished.

broken by SIGUSR1
Processing...
Process finished.

Processing...
broken by SIGUSR1
broken by SIGUSR1
Process finished.

Exiting!
ainfinity@AInfinity:~/ics/lab10$ |
```

