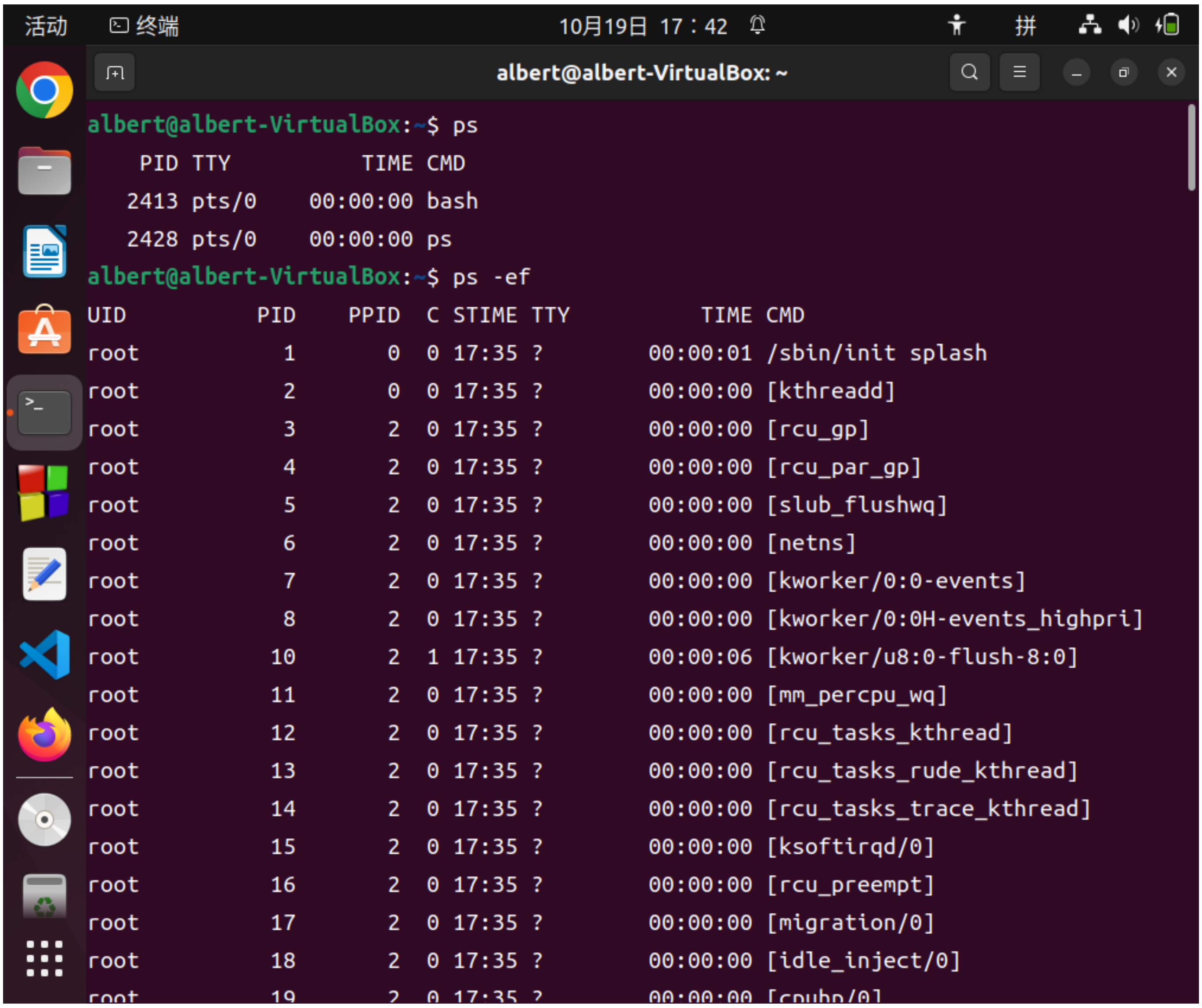
Write a c/c++ program to list all processes in user-mode

Target

1. Write a c/c++ program
2. To list all processes in user-mode

```
ps -ef
```

类似如下结果: UID, PID, PPID, C, STIME, TTY, TIME, CMD



```
albert@albert-VirtualBox:~$ ps
    PID TTY          TIME CMD
   2413 pts/0    00:00:00 bash
   2428 pts/0    00:00:00 ps
albert@albert-VirtualBox:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 17:35 ?        00:00:01 /sbin/init splash
root           2       0  0 17:35 ?        00:00:00 [kthreadd]
root           3       2  0 17:35 ?        00:00:00 [rcu_gp]
root           4       2  0 17:35 ?        00:00:00 [rcu_par_gp]
root           5       2  0 17:35 ?        00:00:00 [slub_flushwq]
root           6       2  0 17:35 ?        00:00:00 [netns]
root           7       2  0 17:35 ?        00:00:00 [kworker/0:0-events]
root           8       2  0 17:35 ?        00:00:00 [kworker/0:0H-events_highpri]
root          10       2  1 17:35 ?        00:00:06 [kworker/u8:0-flush-8:0]
root          11       2  0 17:35 ?        00:00:00 [mm_percpu_wq]
root          12       2  0 17:35 ?        00:00:00 [rcu_tasks_kthread]
root          13       2  0 17:35 ?        00:00:00 [rcu_tasks_rude_kthread]
root          14       2  0 17:35 ?        00:00:00 [rcu_tasks_trace_kthread]
root          15       2  0 17:35 ?        00:00:00 [ksoftirqd/0]
root          16       2  0 17:35 ?        00:00:00 [rcu_preempt]
root          17       2  0 17:35 ?        00:00:00 [migration/0]
root          18       2  0 17:35 ?        00:00:00 [idle_inject/0]
root          19       2  0 17:35 ?        00:00:00 [cpuhp/0]
```

1. GCC
2. IDE 集成开发环境
3. in user-mode
    1. /proc dir contains all informations of all processes

Tools

Install GCC Software Collection

```
sudo apt-get install build-essential
```

How to use GCC

- gcc and make

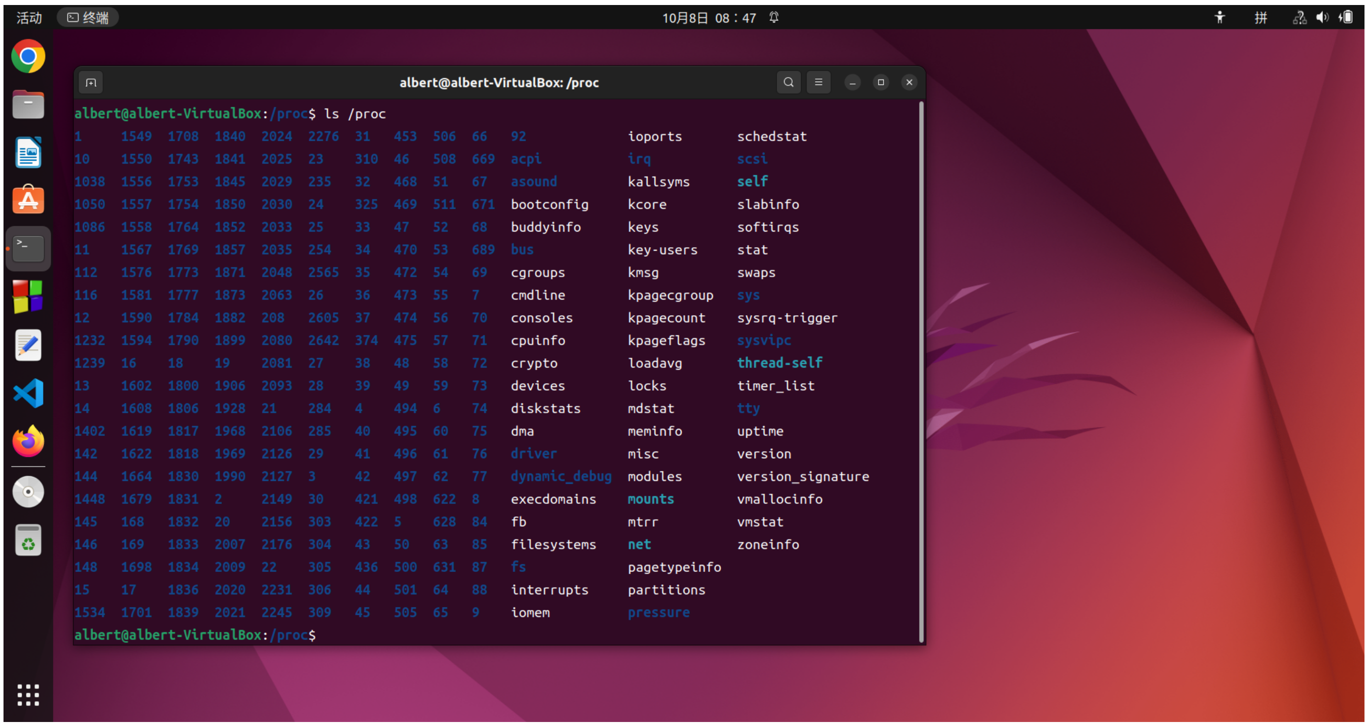
IDE

1. (推荐)Code::Blocks

```
sudo apt-get install codeblocks
```

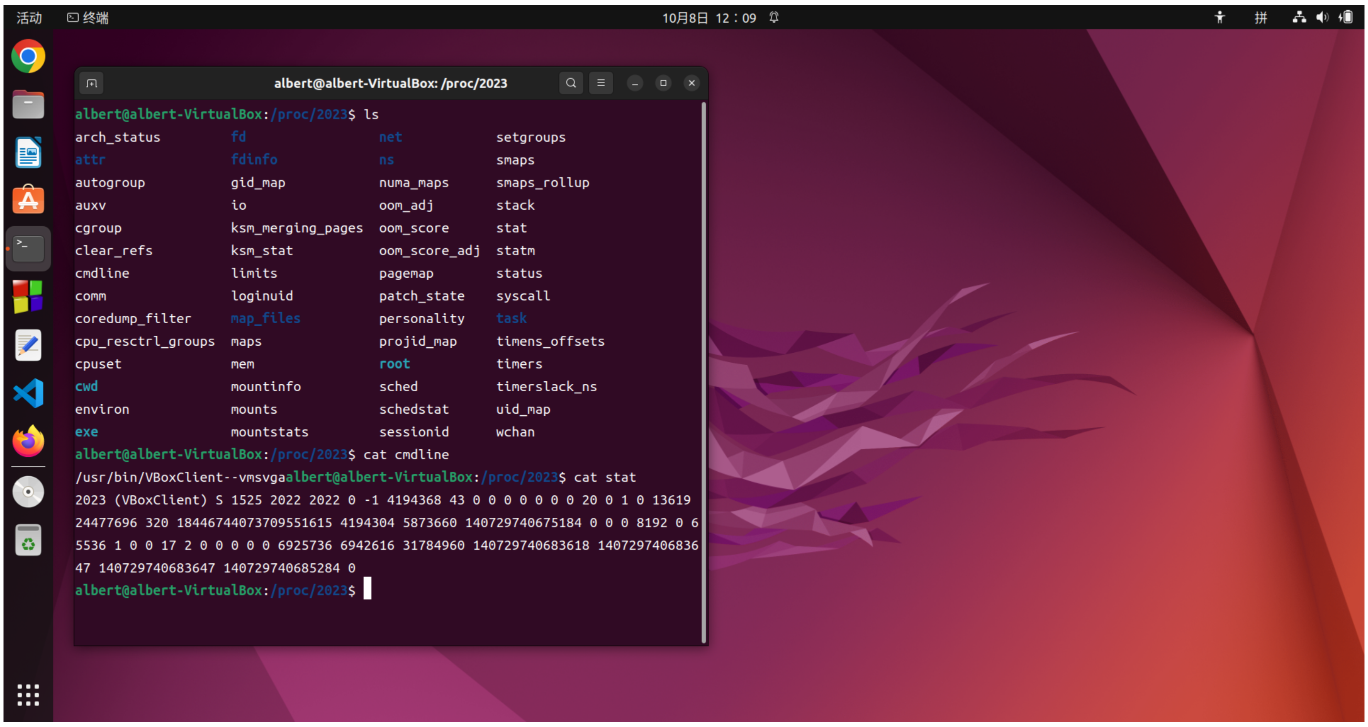
2. vim + gcc

/porc



```
albert@albert-VirtualBox:/proc$ ls /proc
1     1549  1708  1840  2024  2276  31   453  506  66   92             ioports       schedstat
10    1550  1743  1841  2025  23    310  46   508  669  acpi           irq           scsi
1038  1556  1753  1845  2029  235   32   468  51   67   asound         kallsyms      self
1050  1557  1754  1850  2030  24    325  469  511  671  bootconfig     kcore         slabinfo
1086  1558  1764  1852  2033  25    33   47   52   68   buddyinfo      keys          softirqs
11    1567  1769  1857  2035  254   34   470  53   689  bus            key-users     stat
112   1576  1773  1871  2048  2565  35   472  54   69   cgroups        kmsg          swaps
116   1581  1777  1873  2063  26    36   473  55   7    cmdline        kpagecgroup   sys
12    1590  1784  1882  208   2605  37   474  56   70   consoles       kpagecount    sysrq-trigger
1232  1594  1790  1899  2080  2642  374  475  57   71   cpuinfo        kpageflags    sysvipc
1239  16    18    19    2081  27    38   48   58   72   crypto         loadavg       thread-self
13    1602  1800  1906  2093  28    39   49   59   73   devices        locks         timer_list
14    1608  1806  1928  21    284   4    494  6    74   diskstats      mdstat        tty
1402  1619  1817  1968  2106  285   40   495  60   75   dma            meminfo       uptime
142   1622  1818  1969  2126  29    41   496  61   76   driver         misc          version
144   1664  1830  1990  2127  3     42   497  62   77   dynamic_debug  modules       version_signature
1448  1679  1831  2     2149  30    421  498  622  8    execdomains    mounts        vmallocinfo
145   168   1832  20    2156  303   422  5    628  84   fb             mtrr          vmstat
146   169   1833  2007  2176  304   43   50   63   85   filesystems    net           zoneinfo
148   1698  1834  2009  22    305   436  500  631  87   fs             pagetypeinfo
15    17    1836  2020  2231  306   44   501  64   88   interrupts     partitions
1534  1701  1839  2021  2245  309   45   505  65   9    iomem          pressure
albert@albert-VirtualBox:/proc$
```

```
cd /proc/2023
```



```
albert@albert-VirtualBox:/proc/2023$ ls
arch_status         fd                 net            setgroups
attr                fdinfo             ns             smaps
autogroup           gid_map            numa_maps      smaps_rollup
auxv                io                 oom_adj        stack
cgroup              ksm_merging_pages  oom_score      stat
clear_refs          ksm_stat           oom_score_adj  statm
cmdline             limits             pagemap        status
comm                loginuid           patch_state    syscall
coredump_filter     map_files          personality    task
cpu_resctrl_groups  maps               projid_map     timens_offsets
cpuset              mem                root           timers
cwd                 mountinfo          sched          timerslack_ns
environ             mounts             schedstat      uid_map
exe                 mountstats         sessionid      wchan
albert@albert-VirtualBox:/proc/2023$ cat cmdline
/usr/bin/VBoxClient--vmsvgaalbert@albert-VirtualBox:/proc/2023$ cat stat
2023 (VBoxClient) S 1525 2022 2022 0 -1 4194368 43 0 0 0 0 0 0 0 20 0 1 0 13619
24477696 320 18446744073709551615 4194304 5873660 140729740675184 0 0 0 8192 0 6
5536 1 0 0 17 2 0 0 0 0 0 6925736 6942616 31784960 140729740683618 1407297406836
47 140729740683647 140729740685284 0
albert@albert-VirtualBox:/proc/2023$
```

status file:

```
$ more status
Name:   VBoxClient
Umask:  0027
State:  S (sleeping)
Tgid:   2023
Ngid:   0
Pid:    2023
PPid:   1525
TracerPid:  0
Uid:    1000    1000    1000    1000
Gid:    1000    1000    1000    1000
FDSize: 64
Groups: 4 24 27 30 46 122 134 135 999 1000
NStgid: 2023
NSpid:  2023
NSpgid: 2022
NSsid:  2022
VmPeak:    23904 kB
VmSize:    23904 kB
VmLck:         0 kB
VmPin:         0 kB
VmHWM:      1280 kB
VmRSS:      1280 kB
RssAnon:         256 kB
...
```

cmdline file:

```
$cat cmdline
usr/sbin/cups-browsed
$
```

stat file:

```
$cat stat
778 (cups-browsed) S 1 778 778 0 -1 4194560 781 0 22 0 2 0 0 0 20 0 3 0 4249
176762880 2816 18446744073709551615 1 1 0 0 0 0 0 4096 18946 0 0 0 17 1 0 0 0 0 0
0 0 0 0 0 0 0 0
$
```

structure of directory

```
struct dirent
{
    ino_t d_ino; //d_ino 此目录进入点的inode
    ff_t d_off; //d_off 目录文件开头至此目录进入点的位移
    signed short int d_reclen; //d_reclen _name 的长度, 不包含NULL 字符
    unsigned char d_type; //d_type d_name 所指的文件类型 d_name 文件名
    har d_name[256];
};

the value returned in d_type:
              DT_BLK      This is a block device.
              DT_CHR      This is a character device.
              DT_DIR      This is a directory.
              DT_FIFO     This is a named pipe (FIFO).
              DT_LNK      This is a symbolic link.
              DT_REG      This is a regular file.
              DT_SOCK     This is a UNIX domain socket.
              DT_UNKNOWN  The file type could not be determined.

opendir()
readdir()
closedir()
```

Create a symbol link file

```
#include <fcntl.h>           /* Definition of AT_* constants */
#include <unistd.h>
int link(const char *oldpath, const char *newpath);
```

ps

```
$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 12:03 ?        00:00:01 /sbin/init splash
root           2       0  0 12:03 ?        00:00:00 [kthreadd]
root           3       2  0 12:03 ?        00:00:00 [rcu_gp]
root           4       2  0 12:03 ?        00:00:00 [rcu_par_gp]
root           5       2  0 12:03 ?        00:00:00 [slub_flushwq]
root           6       2  0 12:03 ?        00:00:00 [netns]
root           7       2  0 12:03 ?        00:00:00 [kworker/0:0-events]
root           8       2  0 12:03 ?        00:00:00 [kworker/0:0H-events_highpri
root           9       2  0 12:03 ?        00:00:00 [kworker/u8:0-events_unbound
root          10       2  0 12:03 ?        00:00:00 [mm_percpu_wq]
root          11       2  0 12:03 ?        00:00:00 [rcu_tasks_kthread]
root          12       2  0 12:03 ?        00:00:00 [rcu_tasks_rude_kthread]
...
```

How to do

write a c program to list all processes in user-mode, the result is same to ps cmd

1. 列出所有进程，输出结果类似如下命令结果

```
ps -ef
```

...

1. Example of traverse one directory

```
#include <dirent.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
    DIR * dir;
    struct dirent * ptr;
    /*open dir*/
    dir = opendir("/home");
    /*read dir entry*/
    while((ptr = readdir(dir)) != NULL)
    {
        printf("d_name : %s", ptr->d_name);
        if (ptr->d_type==DT_DIR){
            printf("\tDir");
        }
            printf("\n");
    }
    /*close dir*/
    closedir(dir);
    exit(0);
}
```

Compiling:

```
gcc     listdir.c   -o listdir
./listdir
```