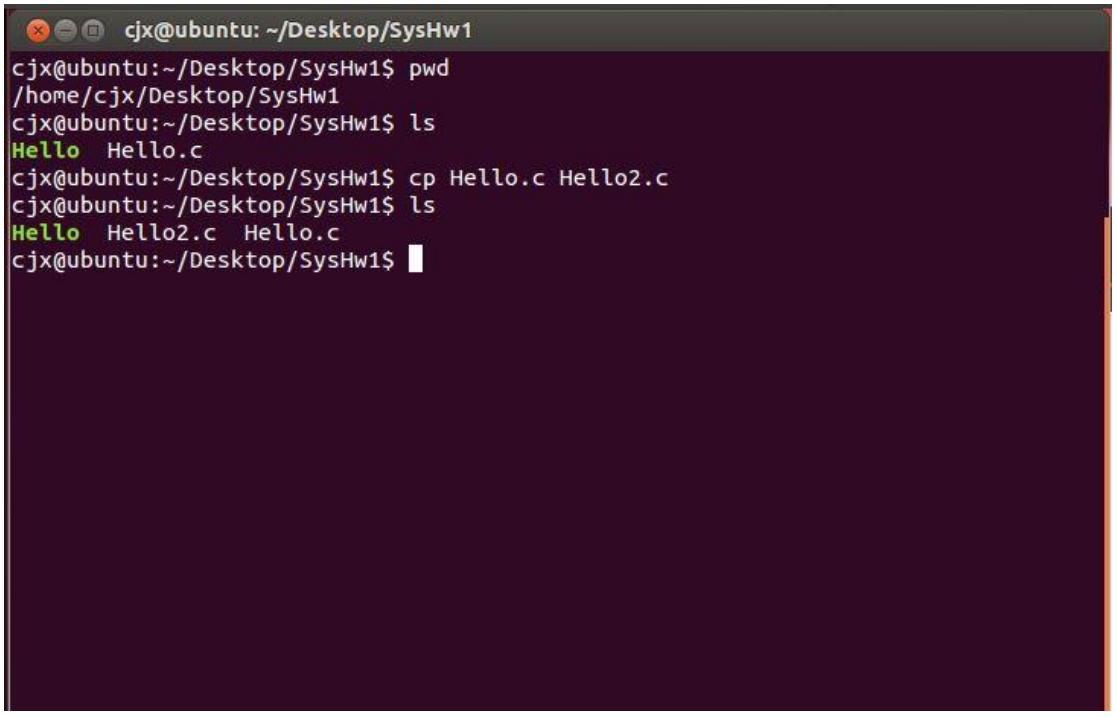
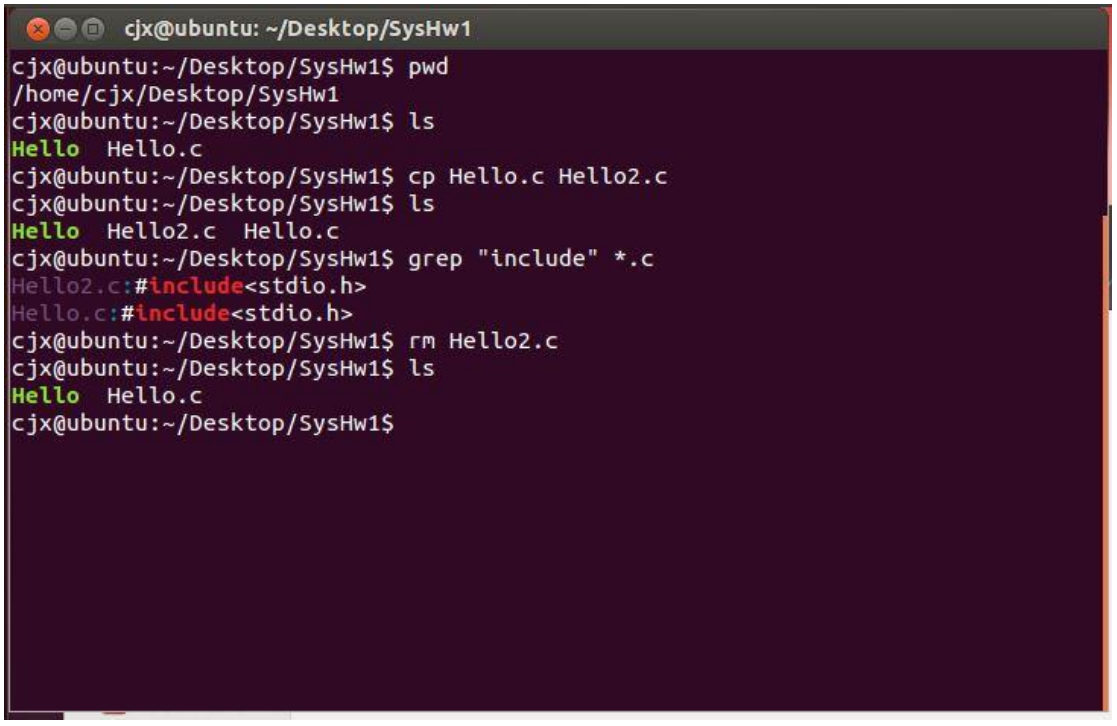
操作系统实验一报告

一、 Linux 的基本指令和 vi 基本编辑

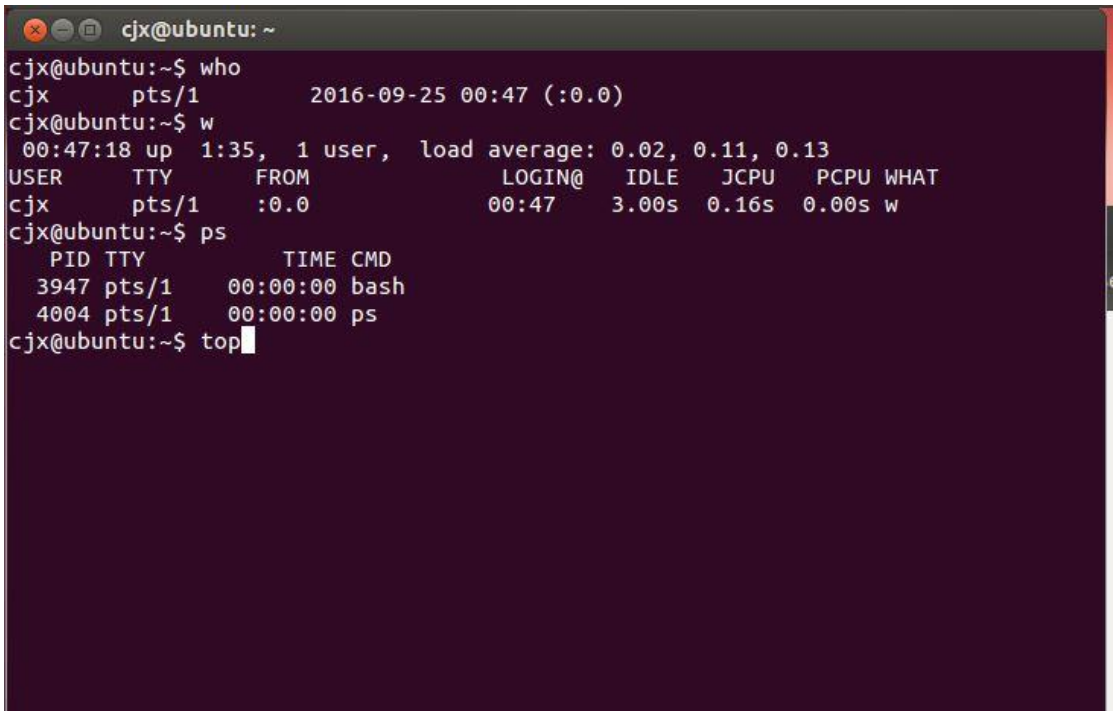
Linux 基本指令有 pwd(查看当前路径)、ls(列举当前目录下的所有文件和目录)、cp(复制文件)、rm(删除文件)、grep（搜索文本等）

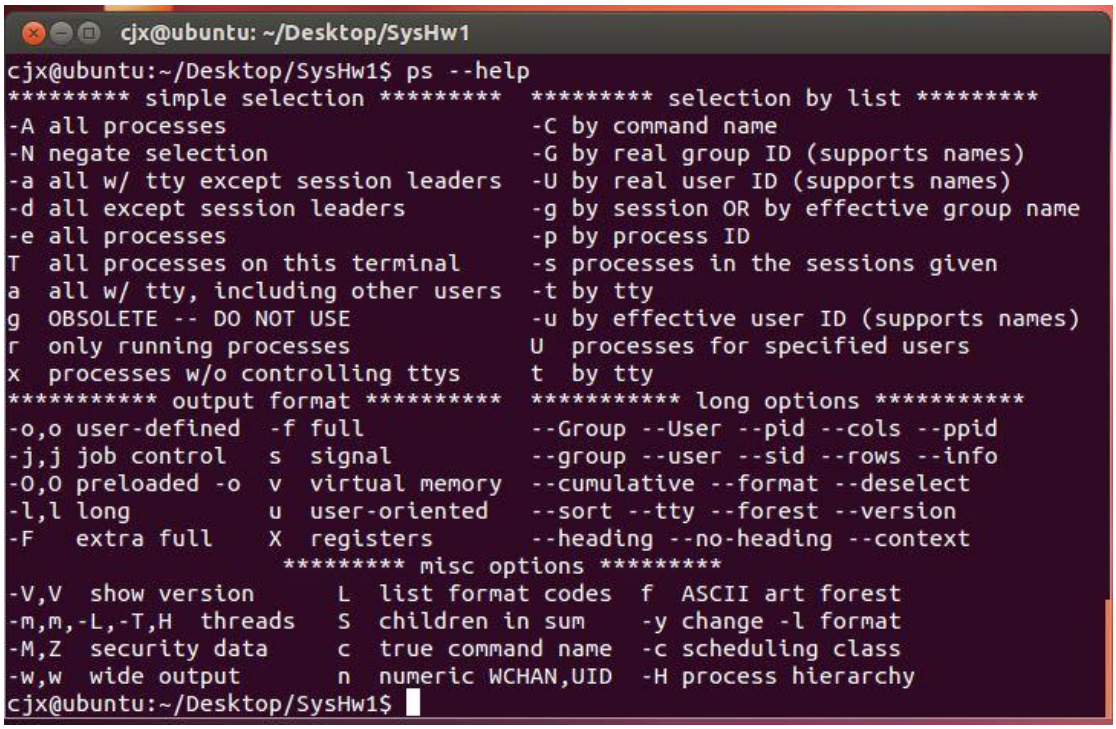


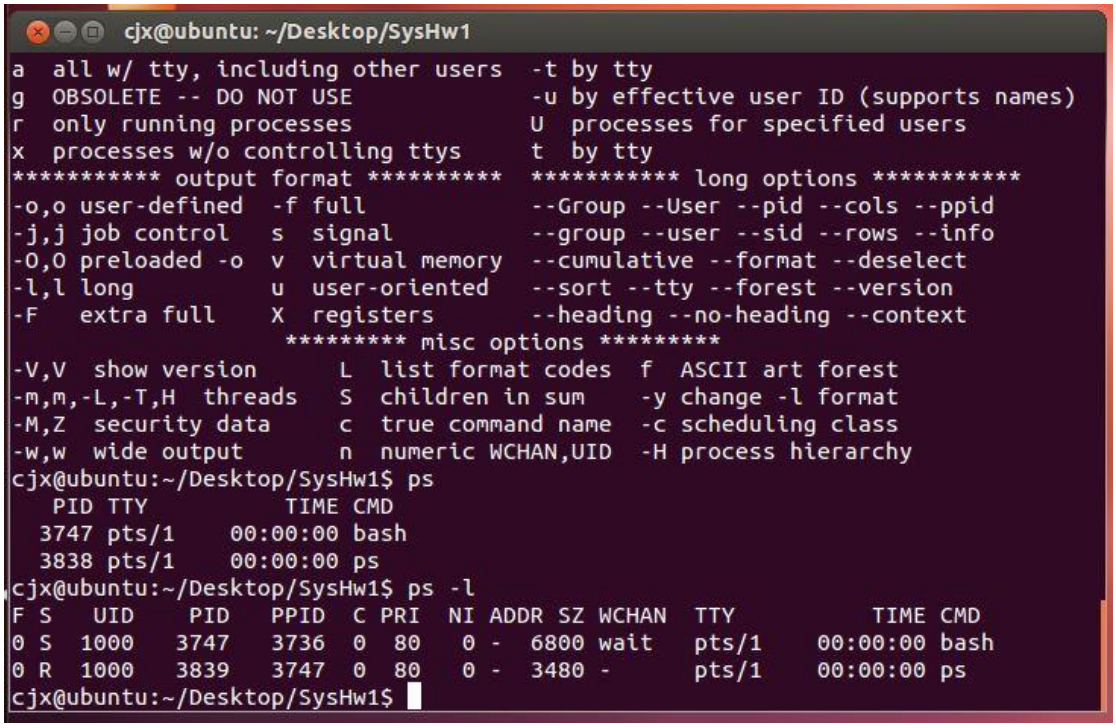


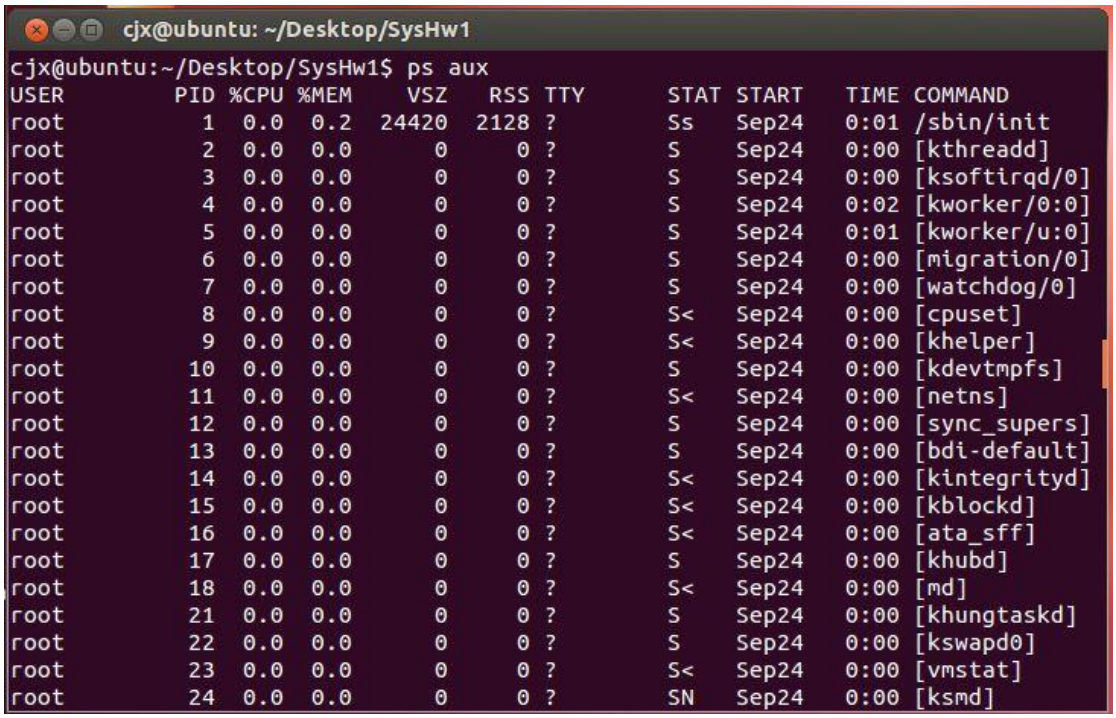
二、 查看 Linux 进程

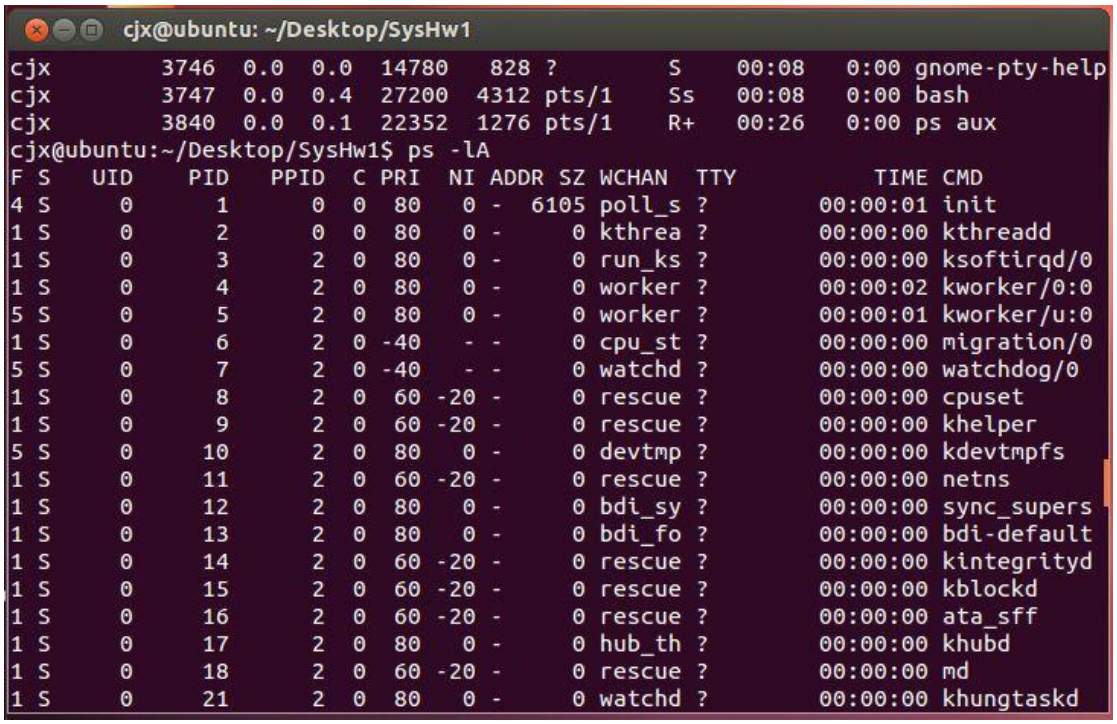
利用 who、w、ps 和 top 等指令查看 Linux 下的进程执行情况

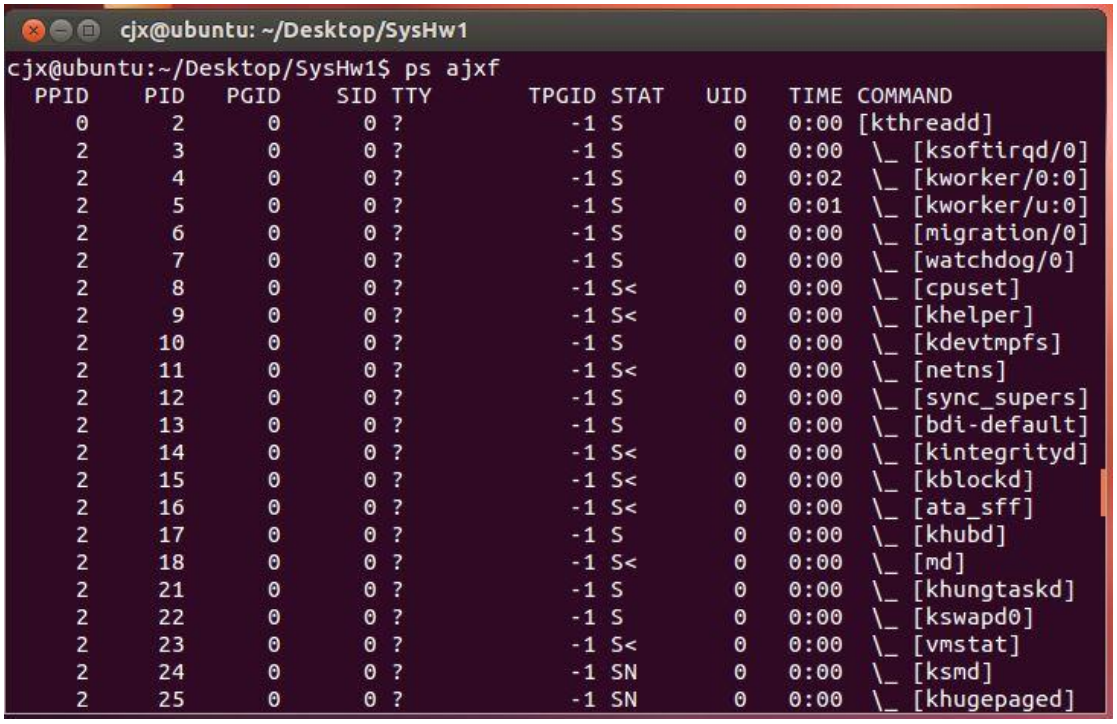


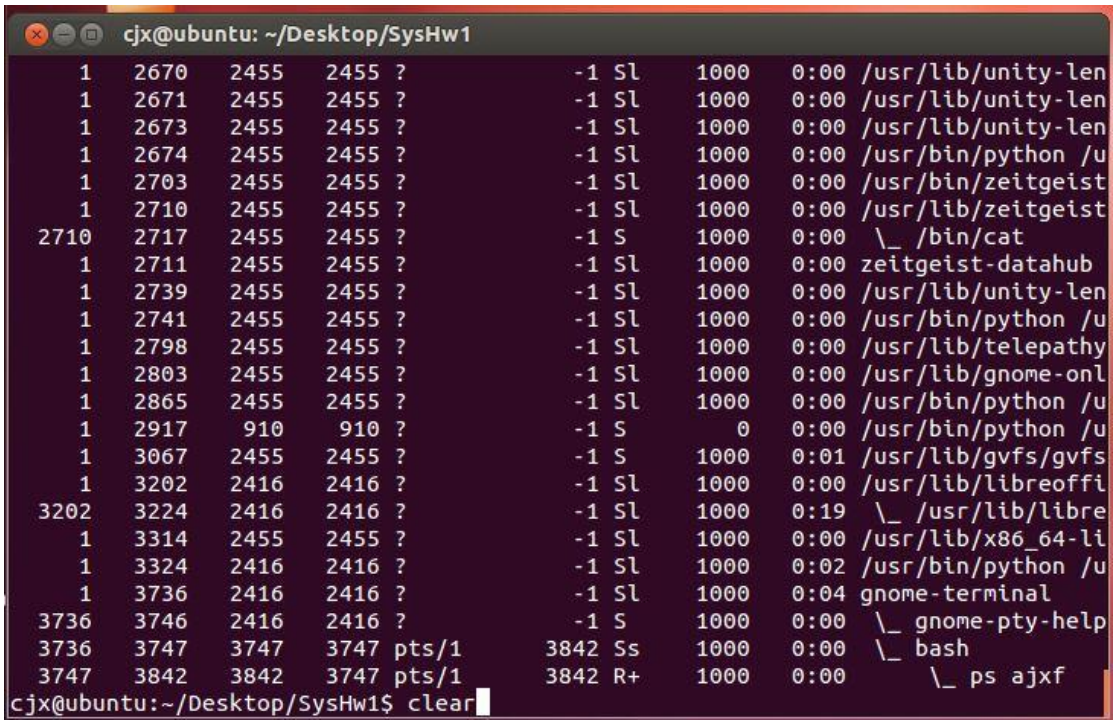



cjx@ubuntu: ~/Desktop/SysHw1
a all w/ tty, including other users -t by tty
g OBSOLETE -- DO NOT USE -u by effective user ID (supports names)
r only running processes U processes for specified users
x processes w/o controlling ttys t by tty
********** output format ********** ********** long options **********
-o,o user-defined -f full --Group --User --pid --cols --ppid
-j,j job control s signal --group --user --sid --rows --info
-O,O preloaded -o v virtual memory --cumulative --format --deselect
-l,l long u user-oriented --sort --tty --forest --version
-F extra full X registers --heading --no-heading --context
******** misc options ********
-V,V show version L list format codes f ASCII art forest
-m,m,-L,-T,H threads S children in sum -y change -l format
-M,Z security data c true command name -c scheduling class
-w,w wide output n numeric WCHAN,UID -H process hierarchy
cjx@ubuntu:~/Desktop/SysHw1$ ps
PID TTY TIME CMD
3747 pts/1 00:00:00 bash
3838 pts/1 00:00:00 ps
cjx@ubuntu:~/Desktop/SysHw1$ ps -l
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 1000 3747 3736 0 80 0 - 6800 wait pts/1 00:00:00 bash
0 R 1000 3839 3747 0 80 0 - 3480 - pts/1 00:00:00 ps
cjx@ubuntu:~/Desktop/SysHw1$


cjx@ubuntu: ~/Desktop/SysHw1
cjx@ubuntu:~/Desktop/SysHw1$ ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.2 24420 2128 ? Ss Sep24 0:01 /sbin/init
root 2 0.0 0.0 0 0 ? S Sep24 0:00 [kthreadd]
root 3 0.0 0.0 0 0 ? S Sep24 0:00 [ksoftirqd/0]
root 4 0.0 0.0 0 0 ? S Sep24 0:02 [kworker/0:0]
root 5 0.0 0.0 0 0 ? S Sep24 0:01 [kworker/u:0]
root 6 0.0 0.0 0 0 ? S Sep24 0:00 [migration/0]
root 7 0.0 0.0 0 0 ? S Sep24 0:00 [watchdog/0]
root 8 0.0 0.0 0 0 ? S< Sep24 0:00 [cpuset]
root 9 0.0 0.0 0 0 ? S< Sep24 0:00 [khelper]
root 10 0.0 0.0 0 0 ? S Sep24 0:00 [kdevtmpfs]
root 11 0.0 0.0 0 0 ? S< Sep24 0:00 [netns]
root 12 0.0 0.0 0 0 ? S Sep24 0:00 [sync_supers]
root 13 0.0 0.0 0 0 ? S Sep24 0:00 [bdi-default]
root 14 0.0 0.0 0 0 ? S< Sep24 0:00 [kintegrityd]
root 15 0.0 0.0 0 0 ? S< Sep24 0:00 [kblockd]
root 16 0.0 0.0 0 0 ? S< Sep24 0:00 [ata_sff]
root 17 0.0 0.0 0 0 ? S Sep24 0:00 [khubd]
root 18 0.0 0.0 0 0 ? S< Sep24 0:00 [md]
root 21 0.0 0.0 0 0 ? S Sep24 0:00 [khungtaskd]
root 22 0.0 0.0 0 0 ? S Sep24 0:00 [kswapd0]
root 23 0.0 0.0 0 0 ? S< Sep24 0:00 [vmstat]
root 24 0.0 0.0 0 0 ? SN Sep24 0:00 [ksmd]

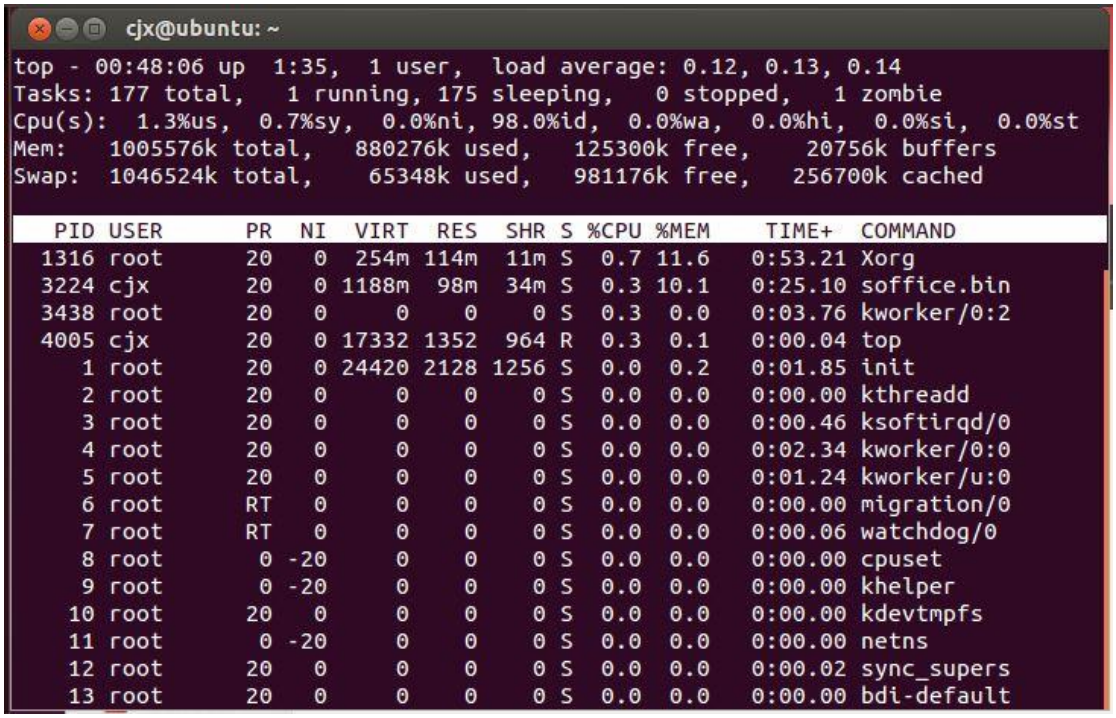

cjx@ubuntu: ~/Desktop/SysHw1
cjx 3746 0.0 0.0 14780 828 ? S 00:08 0:00 gnome-pty-help
cjx 3747 0.0 0.4 27200 4312 pts/1 Ss 00:08 0:00 bash
cjx 3840 0.0 0.1 22352 1276 pts/1 R+ 00:26 0:00 ps aux
cjx@ubuntu:~/Desktop/SysHw1$ ps -lA
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 0 1 0 0 80 0 - 6105 poll_s ? 00:00:01 init
1 S 0 2 0 0 80 0 - 0 kthrea ? 00:00:00 kthreadd
1 S 0 3 2 0 80 0 - 0 run_ks ? 00:00:00 ksoftirqd/0
1 S 0 4 2 0 80 0 - 0 worker ? 00:00:02 kworker/0:0
5 S 0 5 2 0 80 0 - 0 worker ? 00:00:01 kworker/u:0
1 S 0 6 2 0 -40 - - 0 cpu_st ? 00:00:00 migration/0
5 S 0 7 2 0 -40 - - 0 watchd ? 00:00:00 watchdog/0
1 S 0 8 2 0 60 -20 - 0 rescue ? 00:00:00 cpuset
1 S 0 9 2 0 60 -20 - 0 rescue ? 00:00:00 khelper
5 S 0 10 2 0 80 0 - 0 devtmp ? 00:00:00 kdevtmpfs
1 S 0 11 2 0 60 -20 - 0 rescue ? 00:00:00 netns
1 S 0 12 2 0 80 0 - 0 bdi_sy ? 00:00:00 sync_supers
1 S 0 13 2 0 80 0 - 0 bdi_fo ? 00:00:00 bdi-default
1 S 0 14 2 0 60 -20 - 0 rescue ? 00:00:00 kintegrityd
1 S 0 15 2 0 60 -20 - 0 rescue ? 00:00:00 kblockd
1 S 0 16 2 0 60 -20 - 0 rescue ? 00:00:00 ata_sff
1 S 0 17 2 0 80 0 - 0 hub_th ? 00:00:00 khubd
1 S 0 18 2 0 60 -20 - 0 rescue ? 00:00:00 md
1 S 0 21 2 0 80 0 - 0 watchd ? 00:00:00 khungtaskd


cjx@ubuntu: ~/Desktop/SysHw1
cjx@ubuntu:~/Desktop/SysHw1$ ps ajxf
PPID PID PGID SID TTY TPGID STAT UID TIME COMMAND
0 2 0 0 ? -1 S 0 0:00 [kthreadd]
2 3 0 0 ? -1 S 0 0:00 _ [ksoftirqd/0]
2 4 0 0 ? -1 S 0 0:02 _ [kworker/0:0]
2 5 0 0 ? -1 S 0 0:01 _ [kworker/u:0]
2 6 0 0 ? -1 S 0 0:00 _ [migration/0]
2 7 0 0 ? -1 S 0 0:00 _ [watchdog/0]
2 8 0 0 ? -1 S< 0 0:00 _ [cpuset]
2 9 0 0 ? -1 S< 0 0:00 _ [khelper]
2 10 0 0 ? -1 S 0 0:00 _ [kdevtmpfs]
2 11 0 0 ? -1 S< 0 0:00 _ [netns]
2 12 0 0 ? -1 S 0 0:00 _ [sync_supers]
2 13 0 0 ? -1 S 0 0:00 _ [bdi-default]
2 14 0 0 ? -1 S< 0 0:00 _ [kintegrityd]
2 15 0 0 ? -1 S< 0 0:00 _ [kblockd]
2 16 0 0 ? -1 S< 0 0:00 _ [ata_sff]
2 17 0 0 ? -1 S 0 0:00 _ [khubd]
2 18 0 0 ? -1 S< 0 0:00 _ [md]
2 21 0 0 ? -1 S 0 0:00 _ [khungtaskd]
2 22 0 0 ? -1 S 0 0:00 _ [kswapd0]
2 23 0 0 ? -1 S< 0 0:00 _ [vmstat]
2 24 0 0 ? -1 SN 0 0:00 _ [ksmd]
2 25 0 0 ? -1 SN 0 0:00 _ [khugepaged]



三、

四、 Linux 进程信息（写出具体含义）



F: 标志位（flags），具体有哪些值及其含义可以参考 man ps 里面关于 flags 的相关内容

S: 状态，下面那几个进程的状态为 S 表示是在 sleeping 状态（即等待状态）

PID: 当前进程号
PPID: 当前进程的父进程号
C: CPU 占用百分率
PRI:优先级
NI: Nice 值
ADDR: 交换地址

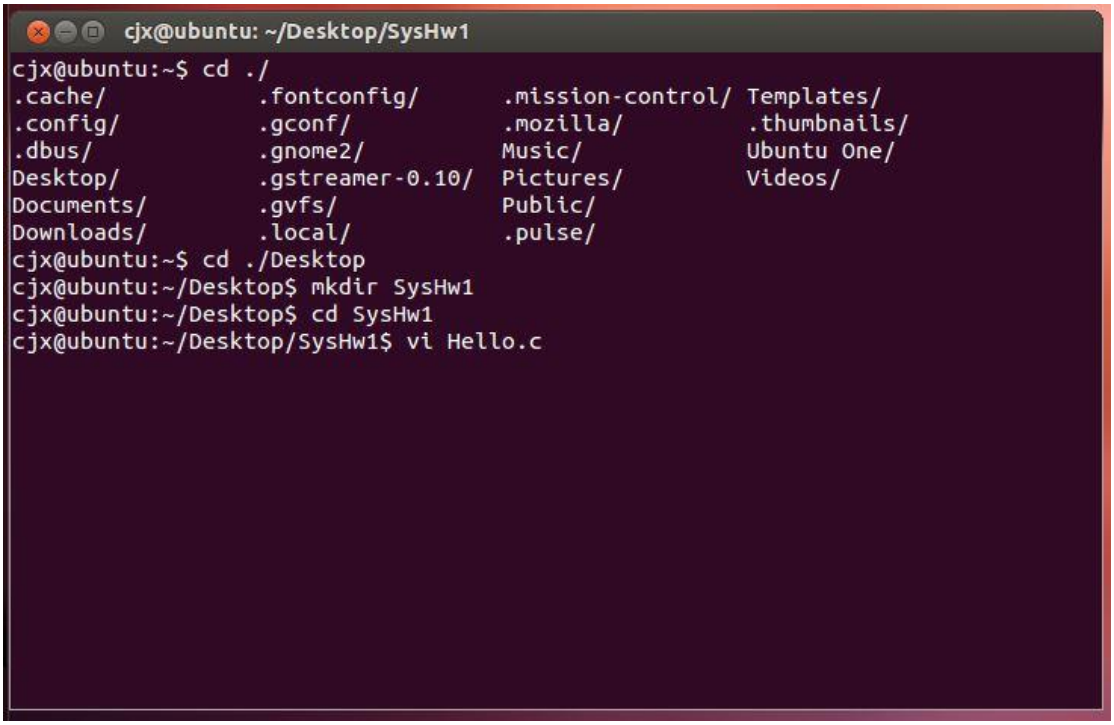
SZ:虚拟内存大小
WCHAN: 当前进程正在等待的事件（比如等待 semphore，等待 IO 等）
STIME: 进程启动时间 （什么时候开始运行的）
TIME: 进程占用 cpu 时间
TTY: 进程对应控制终端 （可以没有）
CMD: 进程对应的命令行参数

五、 Linux 使用体会（与 Window 相比）

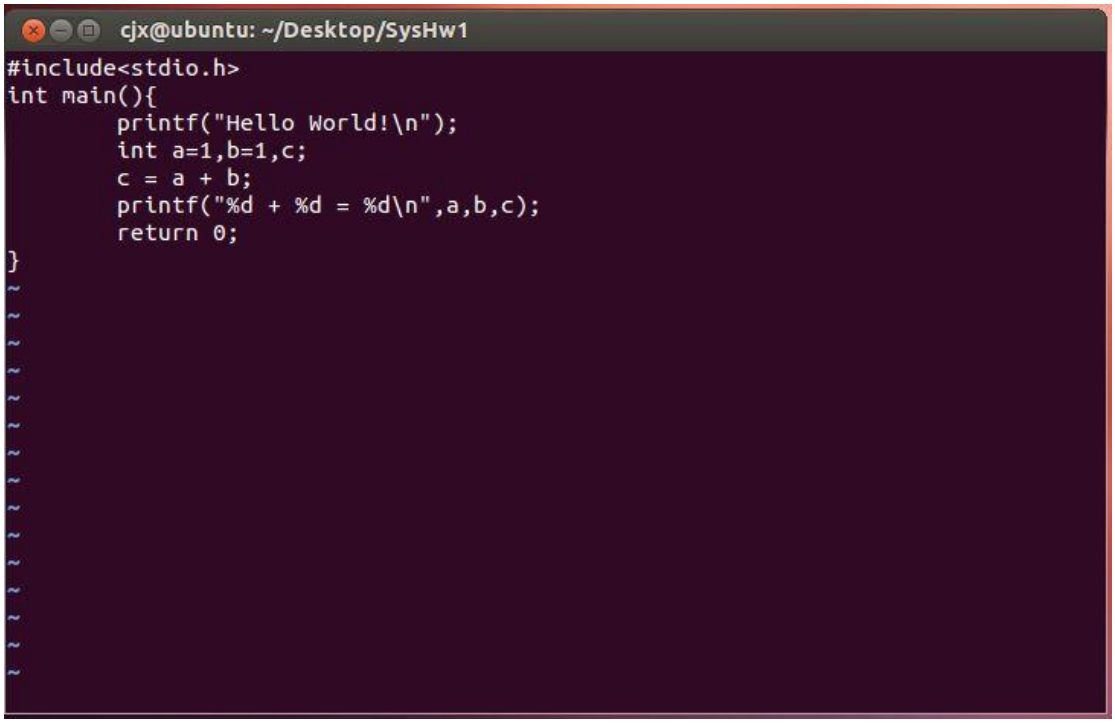
初次使用 linux 系统，通过终端敲击一条条指令进行系统操作，相比于 window 的图形界面，确实有些不习惯，用得不是太方便。但是通过一条条指令的操作，我更加清楚地了解到了计算机运行的原理。比如 C 语言程序的编辑和运行，先是 vi 编辑，再 gcc，最后 gdb，程序运行背后的原理就显得具体形象，而不像 window 那样抽象成几个图形按键。通过网上查找资料得知 linux 是编程的最后系统，我似乎有那么一点点理解，记下来我需要继续使用 linux 系统，才能发现更多和应用 linux 系统的“过机之处”。

六、 vi、gcc 和 gdc 的使用

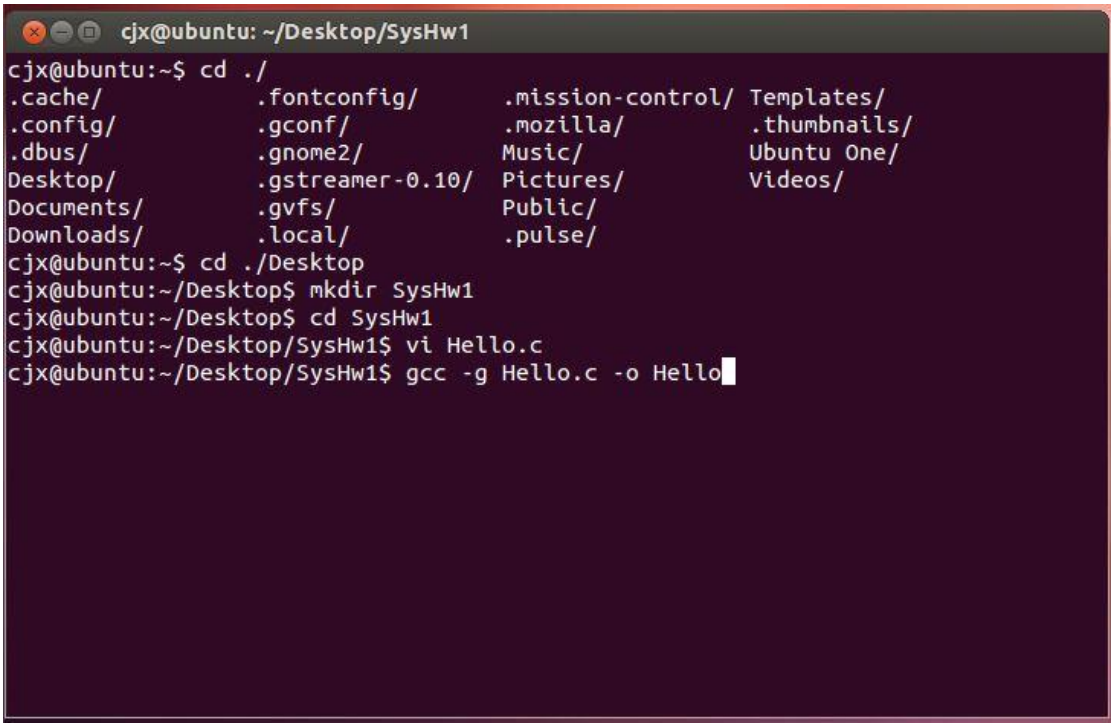
1. 创建文件



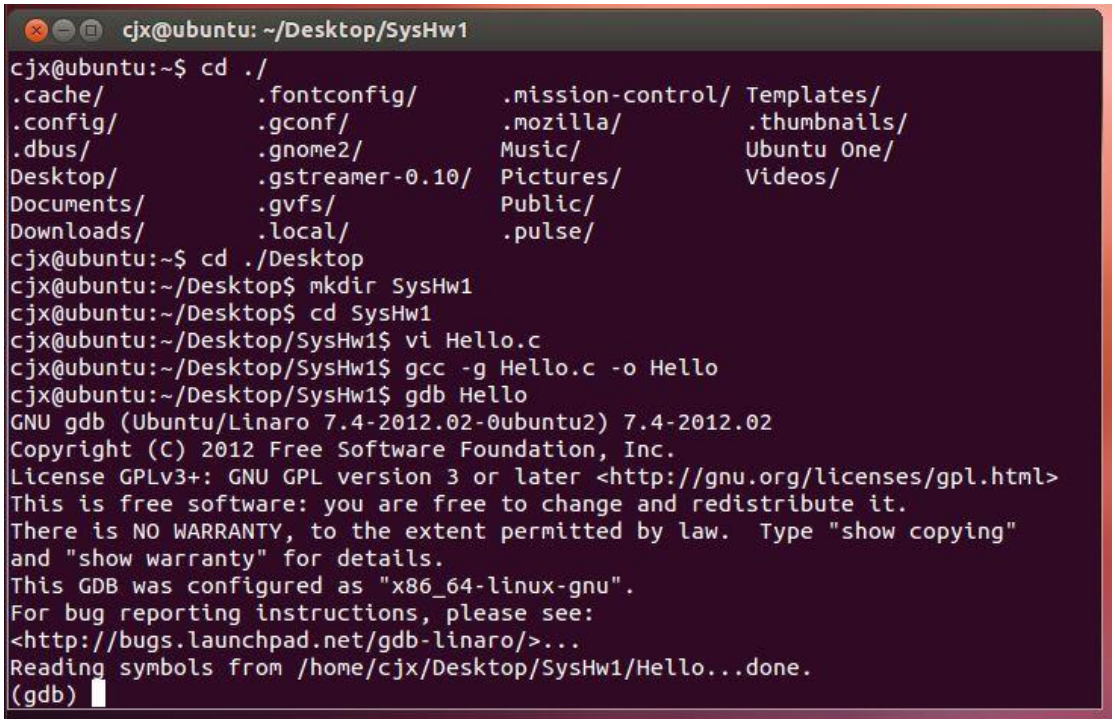
2. Vi 编辑文件



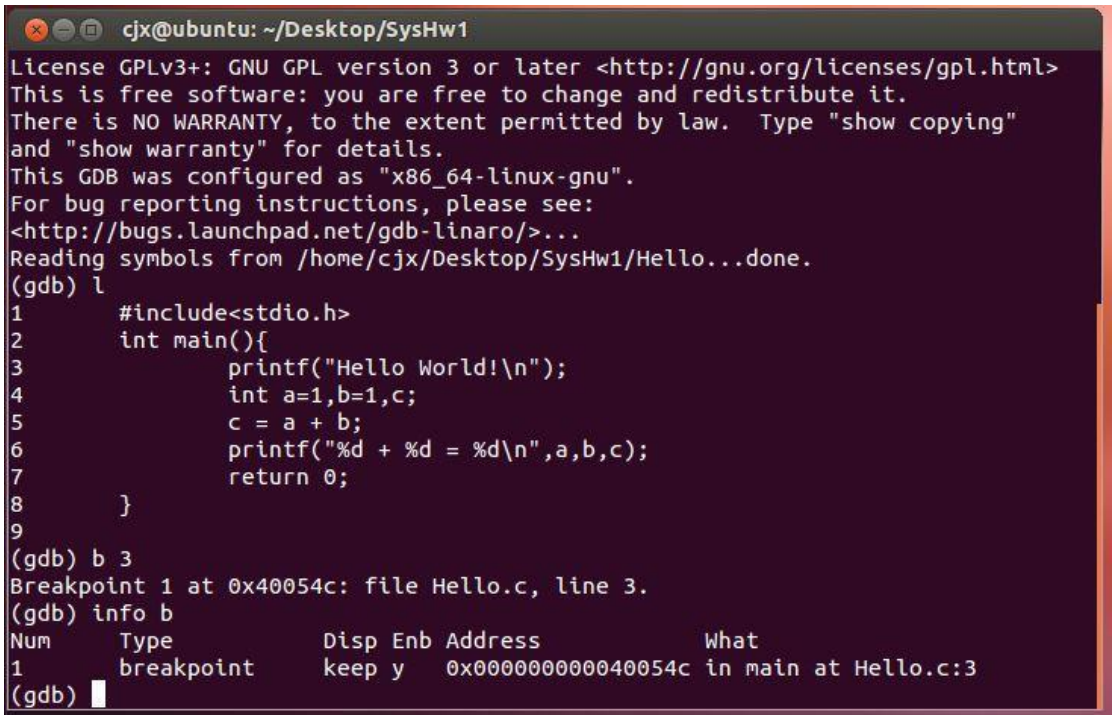
3. Gcc -g Hello.c -o Hello



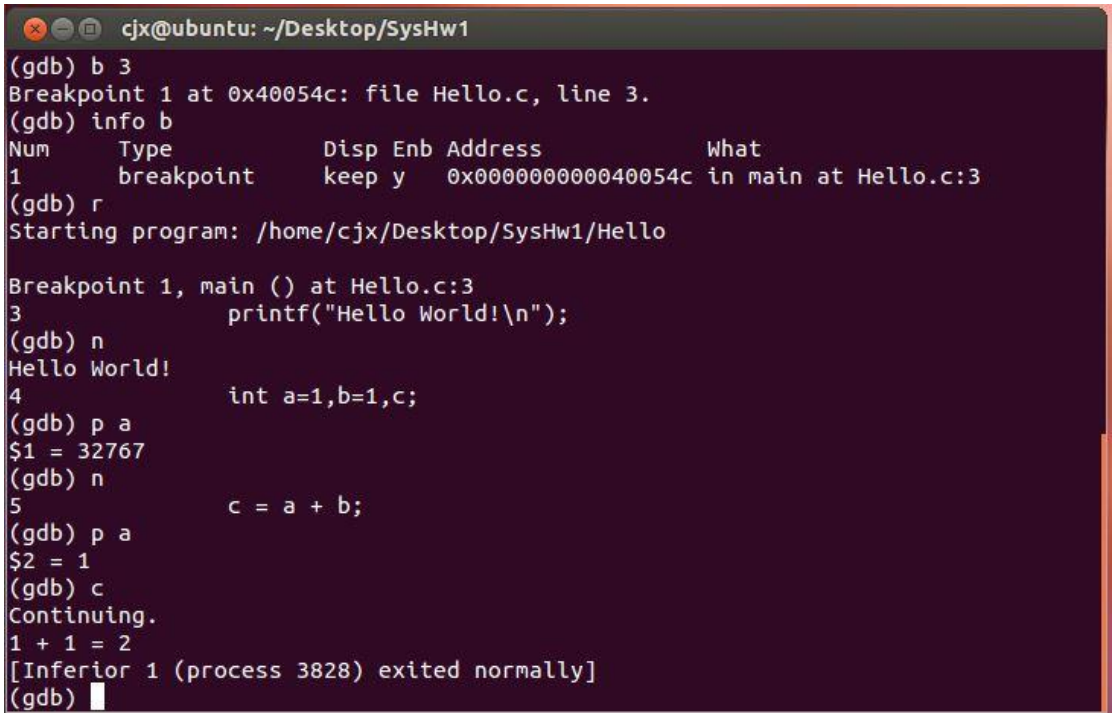
4. Gdb Hello



5. l 查看载入文件



6. b 3 设置断点



7. info b 查看断点信息
8. r 运行至断点
9. p a 查看变量 a 的值
10. n 下一步
11. c 继续运行