BSM 308-System Programming

Sakarya University-Computer Engineering

Contents

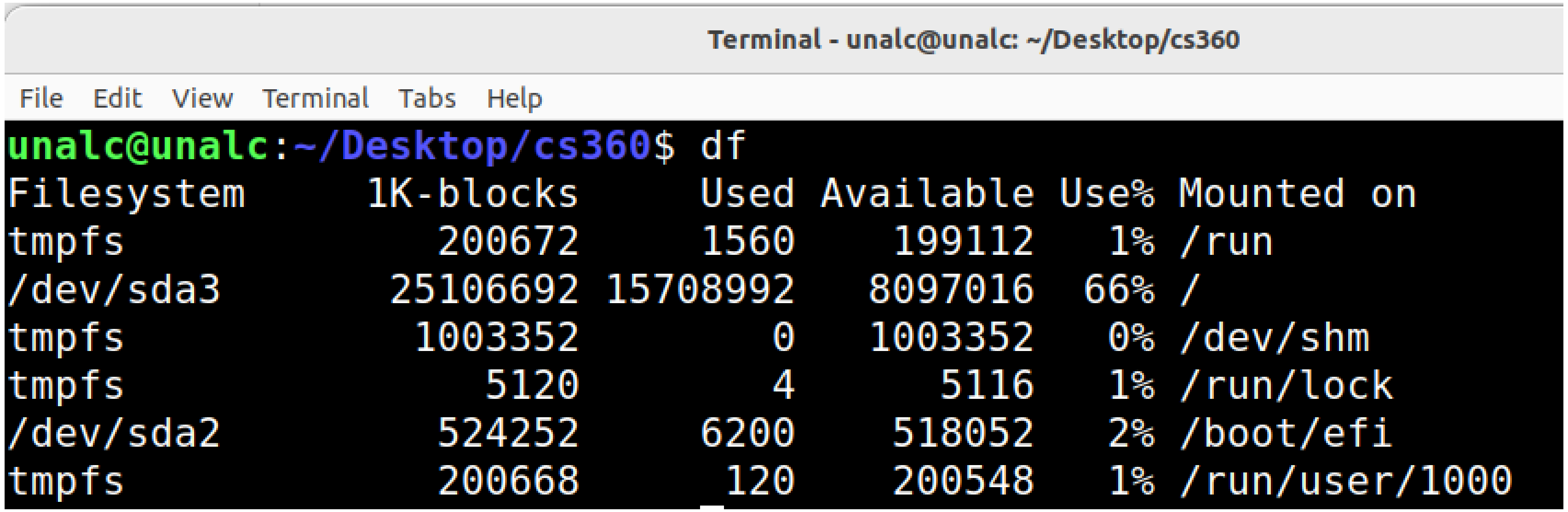
a) Some Basic Terminology

b) Introduction to Signals App.

- http://web.eecs.utk.edu/~jplank/plank/classes/cs360/360/notes/Chap1/lecture.html
- http://web.eecs.utk.edu/~jplank/plank/classes/cs360/360/notes/Syscall-Intro/lecture.html

File System

- Definition of a ``filesystem'': A hierarchical arrangement of directories.
- In Unix, the root filesystem starts with "/". However, there are other subfilesystems, that are part of the root one. To see the filesystems on your machine, type "df".


Terminal - unalc@unalc: ~/Desktop/cs360
File Edit View Terminal Tabs Help

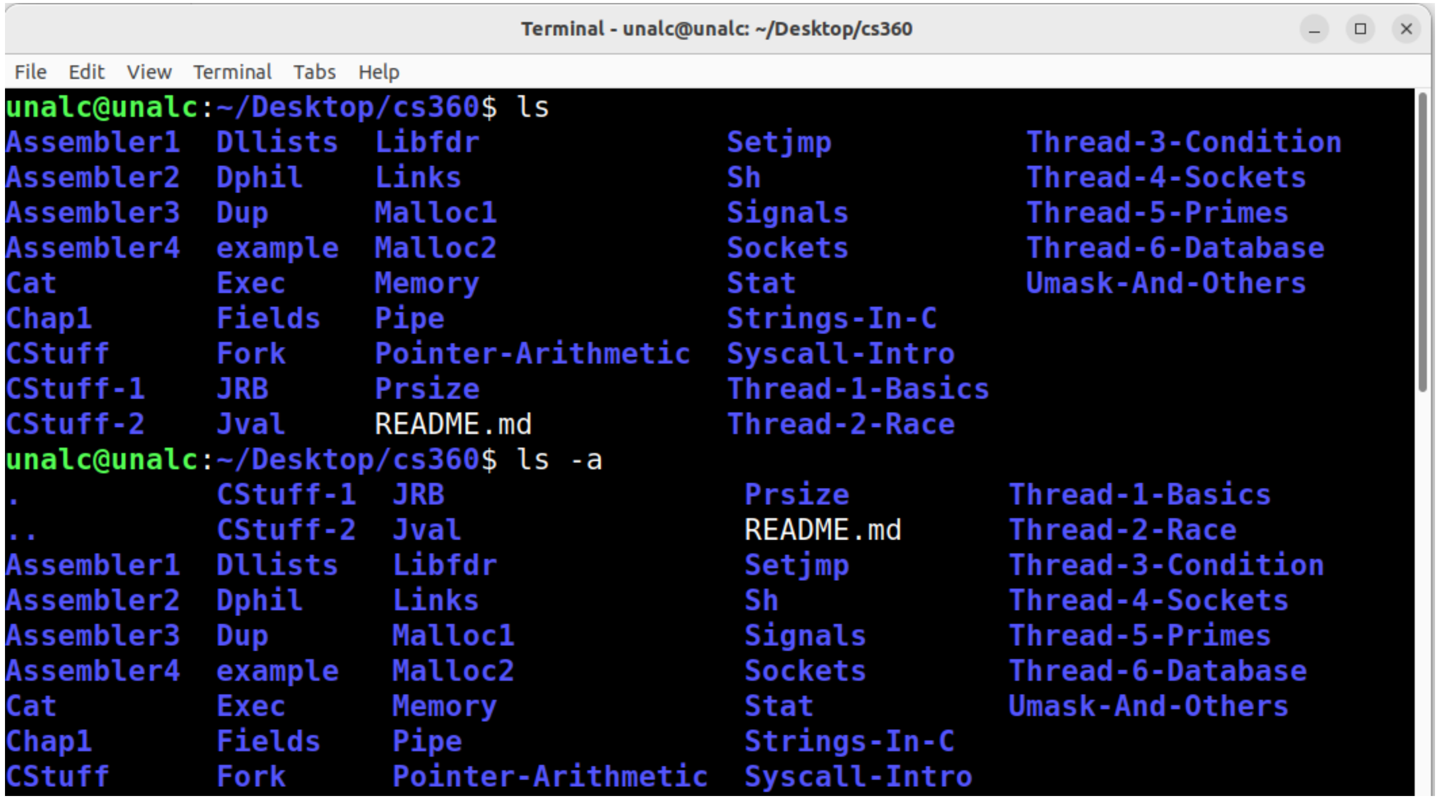
```
unalc@unalc:~/Desktop/cs360$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs             200672     1560    199112   1% /run
/dev/sda3       25106692 15708992   8097016  66% /
tmpfs            1003352        0   1003352   0% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
/dev/sda2         524252     6200    518052   2% /boot/efi
tmpfs             200668      120    200548   1% /run/user/1000
```

File System

- ❑Each line represents a different file system.
- ❑The first entry in the line indicates where the file system is located, and the last entry indicates how you accessed it on your machine.
- ❑For example, the file system /dev/sda1 is a partition of one of the disks on the machine. I can get it from **/** (root directory).
- ❑The way file systems work varies from year to year, but you can usually parse the output of df and figure out what's going on.

Names, Paths, Directories

- Filename : The name of a file as it appears in a directory.
- Pathname : A sequence of zero or more filenames separated by forward slashes .
- " ls -a" it lists all filenames in the current directory:
- **"." is the current directory. ".." is the parent of the current directory.**

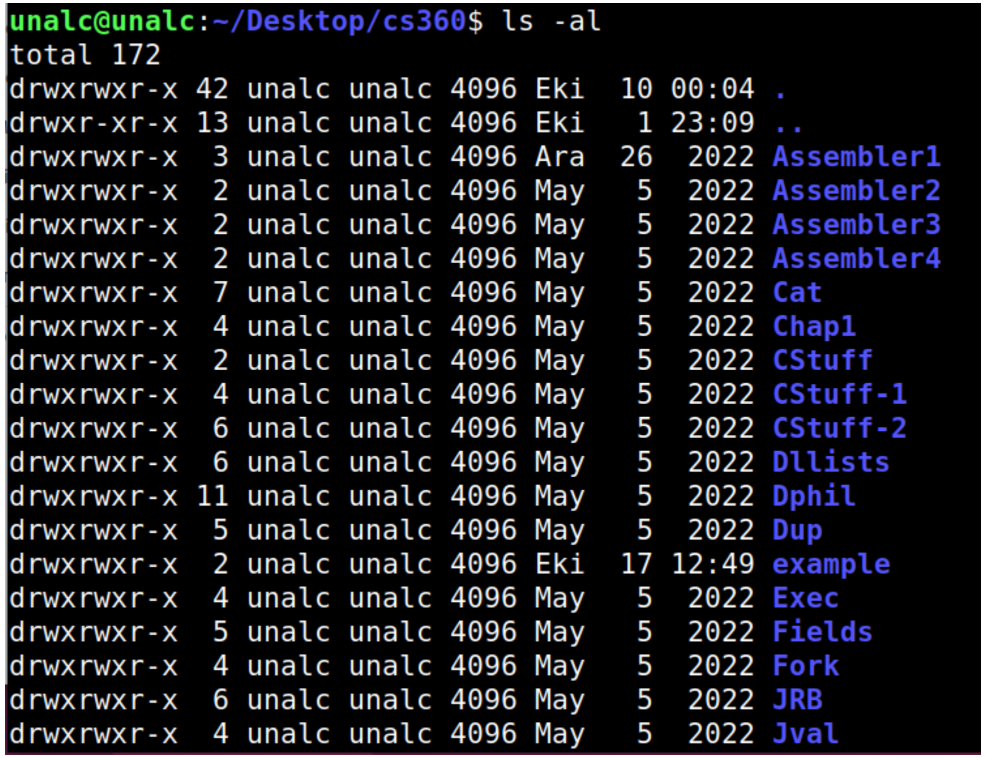


```
unalc@unalc:~/Desktop/cs360$ ls
Assembler1  Dllists  Libfdr              Setjmp           Thread-3-Condition
Assembler2  Dphil    Links               Sh               Thread-4-Sockets
Assembler3  Dup      Malloc1             Signals          Thread-5-Primes
Assembler4  example  Malloc2             Sockets          Thread-6-Database
Cat         Exec     Memory              Stat             Umask-And-Others
Chap1       Fields   Pipe                Strings-In-C
CStuff      Fork     Pointer-Arithmetic  Syscall-Intro
CStuff-1    JRB      Prsize              Thread-1-Basics
CStuff-2    Jval     README.md           Thread-2-Race
unalc@unalc:~/Desktop/cs360$ ls -a
.           CStuff-1  JRB                 Prsize         Thread-1-Basics
..          CStuff-2  Jval                README.md      Thread-2-Race
Assembler1  Dllists   Libfdr              Setjmp         Thread-3-Condition
Assembler2  Dphil     Links               Sh             Thread-4-Sockets
Assembler3  Dup       Malloc1             Signals        Thread-5-Primes
Assembler4  example   Malloc2             Sockets        Thread-6-Database
Cat         Exec      Memory              Stat           Umask-And-Others
Chap1       Fields    Pipe                Strings-In-C
CStuff      Fork      Pointer-Arithmetic  Syscall-Intro
```

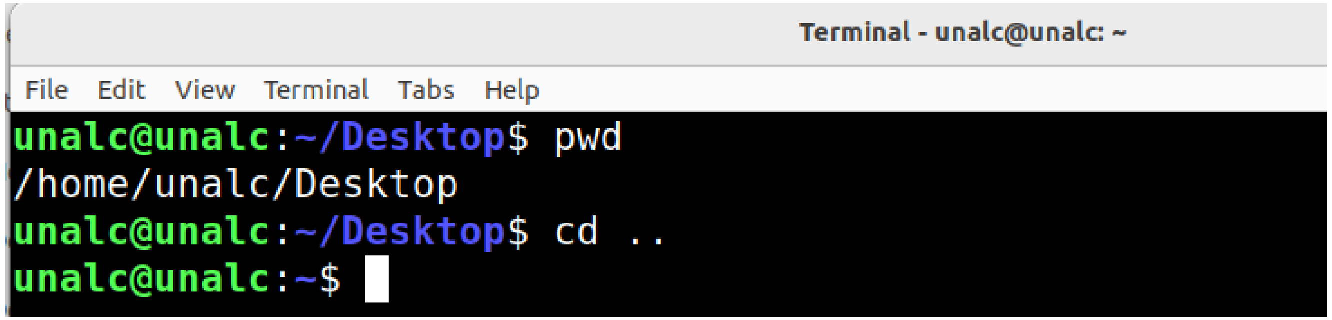
Names, Paths,Directories

- The pwd command tells you the full pathname of the current directory.

- The cd command moves you between directories:



```
unalc@unalc:~/Desktop/cs360$ ls -al
total 172
drwxrwxr-x 42 unalc unalc 4096 Eki  10 00:04 .
drwxr-xr-x 13 unalc unalc 4096 Eki   1 23:09 ..
drwxrwxr-x  3 unalc unalc 4096 Ara  26  2022 Assembler1
drwxrwxr-x  2 unalc unalc 4096 May   5  2022 Assembler2
drwxrwxr-x  2 unalc unalc 4096 May   5  2022 Assembler3
drwxrwxr-x  2 unalc unalc 4096 May   5  2022 Assembler4
drwxrwxr-x  7 unalc unalc 4096 May   5  2022 Cat
drwxrwxr-x  4 unalc unalc 4096 May   5  2022 Chap1
drwxrwxr-x  2 unalc unalc 4096 May   5  2022 CStuff
drwxrwxr-x  4 unalc unalc 4096 May   5  2022 CStuff-1
drwxrwxr-x  6 unalc unalc 4096 May   5  2022 CStuff-2
drwxrwxr-x  6 unalc unalc 4096 May   5  2022 Dllists
drwxrwxr-x 11 unalc unalc 4096 May   5  2022 Dphil
drwxrwxr-x  5 unalc unalc 4096 May   5  2022 Dup
drwxrwxr-x  2 unalc unalc 4096 Eki  17 12:49 example
drwxrwxr-x  4 unalc unalc 4096 May   5  2022 Exec
drwxrwxr-x  5 unalc unalc 4096 May   5  2022 Fields
drwxrwxr-x  4 unalc unalc 4096 May   5  2022 Fork
drwxrwxr-x  6 unalc unalc 4096 May   5  2022 JRB
drwxrwxr-x  4 unalc unalc 4096 May   5  2022 Jval
```



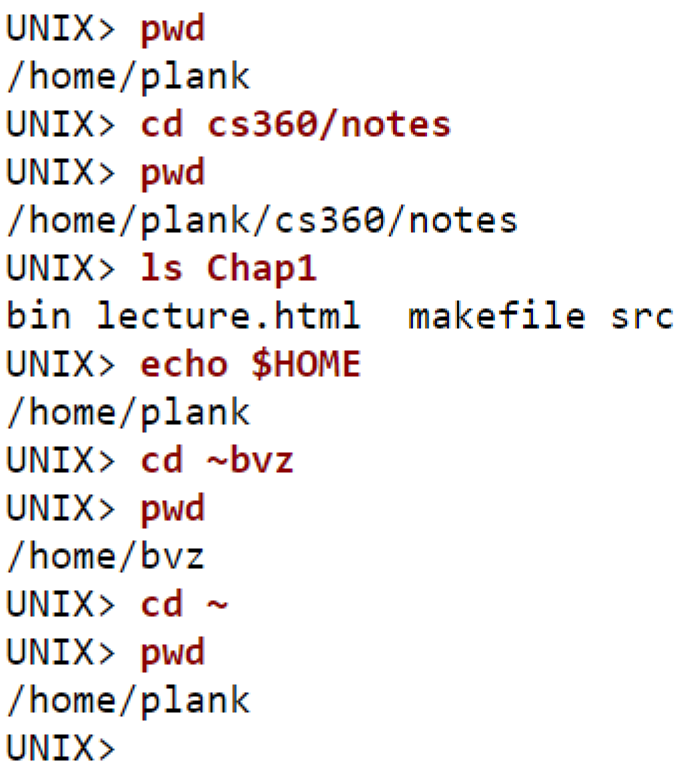
```
unalc@unalc:~/Desktop$ pwd
/home/unalc/Desktop
unalc@unalc:~/Desktop$ cd ..
unalc@unalc:~$ 
```

Names, Paths, Directories

- Absolute Path Name: A path name that begins with a slash.
- Relative Path Name: A path name that does not begin with a slash.
- Working Directory: The directory to which relative pathnames are relative.

your working directory with pwd .

- Home Directory: A user's working directory when they first log in.



```
UNIX> pwd
/home/plank
UNIX> cd cs360/notes
UNIX> pwd
/home/plank/cs360/notes
UNIX> ls Chap1
bin lecture.html  makefile src
UNIX> echo $HOME
/home/plank
UNIX> cd ~bvz
UNIX> pwd
/home/bvz
UNIX> cd ~
UNIX> pwd
/home/plank
UNIX>
```

Programs and processes:

- **Program**: *A file that can be executed, either directly, or through the aid of interpreters, compilers, and/or linkers.*
- **Process**: *An executing instance of a program.*
- **Process ID**: *The number given to a process by the operating system.*

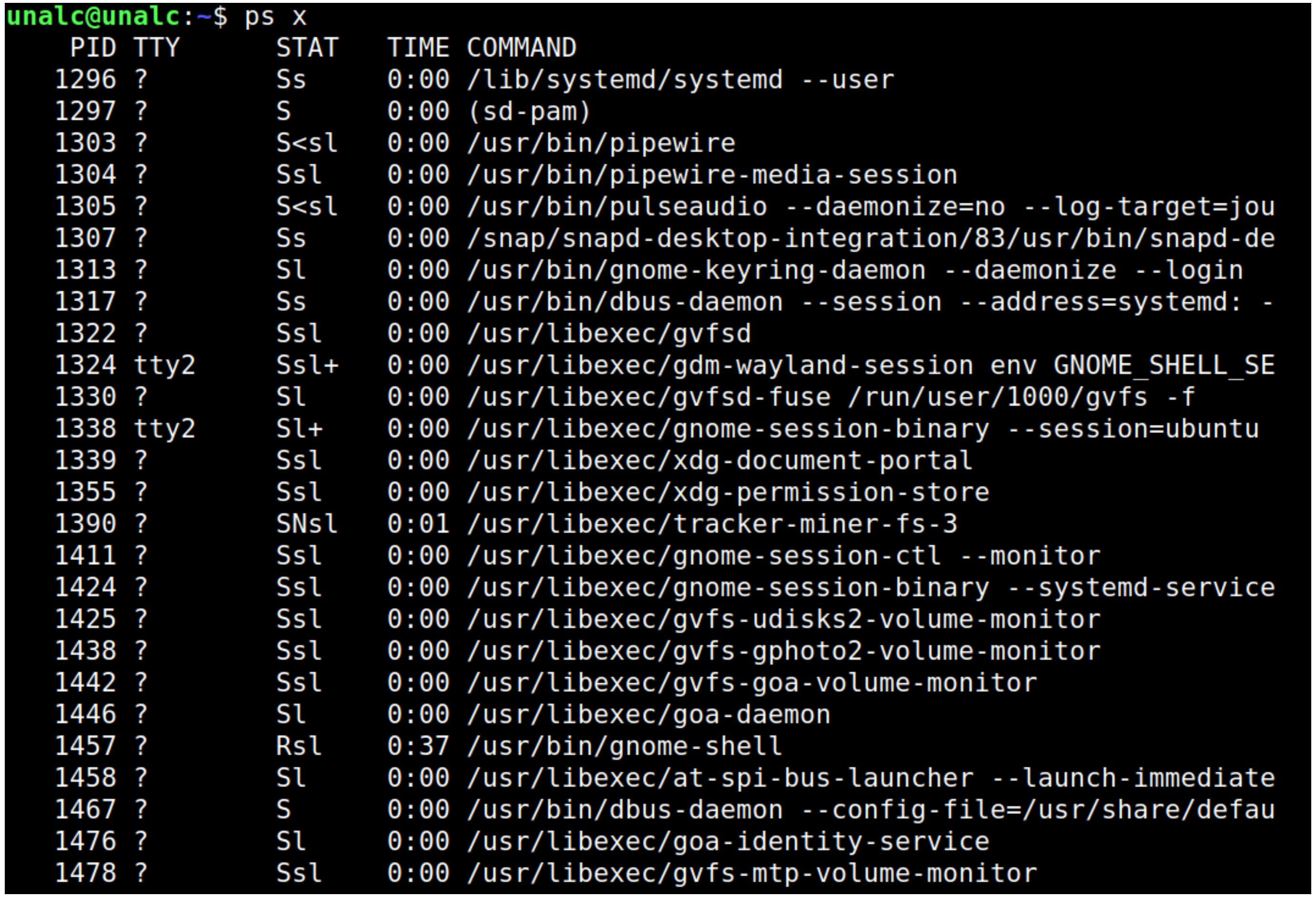
Examples of programs:

- **/bin/ls** -- this is a program that can be executed directly.
- **/usr/bin/vim** -- this is a program that can be executed directly.
- **/home/plank/cs360/notes/Chap1/src/ch1a.c** -- this is a program that needs to be compiled in order to execute it.
- **/home/plank/bin/calc** -- this is a shell script -- it is a program that needs to be interpreted by **/bin/sh**.

When you run a program, its executing instance is called a *process*.

Processes

- " ps x " will list all the processes you are currently executing:

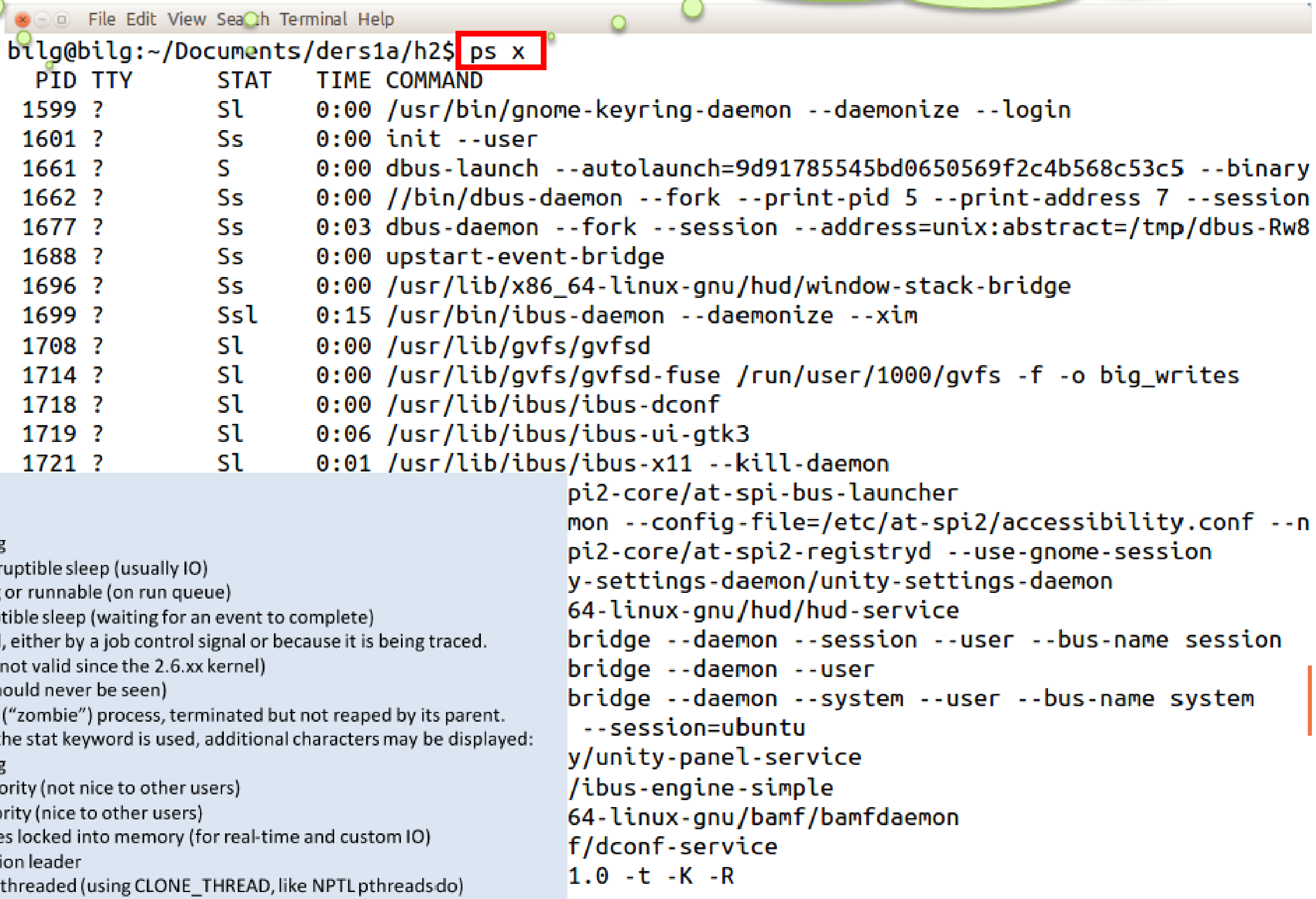


```
unalc@unalc:~$ ps x
    PID TTY      STAT   TIME COMMAND
   1296 ?        Ss     0:00 /lib/systemd/systemd --user
   1297 ?        S      0:00 (sd-pam)
   1303 ?        S<sl   0:00 /usr/bin/pipewire
   1304 ?        Ssl    0:00 /usr/bin/pipewire-media-session
   1305 ?        S<sl   0:00 /usr/bin/pulseaudio --daemonize=no --log-target=jou
   1307 ?        Ss     0:00 /snap/snapd-desktop-integration/83/usr/bin/snapd-de
   1313 ?        Sl     0:00 /usr/bin/gnome-keyring-daemon --daemonize --login
   1317 ?        Ss     0:00 /usr/bin/dbus-daemon --session --address=systemd: -
   1322 ?        Ssl    0:00 /usr/libexec/gvfsd
   1324 tty2     Ssl+   0:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SE
   1330 ?        Sl     0:00 /usr/libexec/gvfsd-fuse /run/user/1000/gvfs -f
   1338 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --session=ubuntu
   1339 ?        Ssl    0:00 /usr/libexec/xdg-document-portal
   1355 ?        Ssl    0:00 /usr/libexec/xdg-permission-store
   1390 ?        SNsl   0:01 /usr/libexec/tracker-miner-fs-3
   1411 ?        Ssl    0:00 /usr/libexec/gnome-session-ctl --monitor
   1424 ?        Ssl    0:00 /usr/libexec/gnome-session-binary --systemd-service
   1425 ?        Ssl    0:00 /usr/libexec/gvfs-udisks2-volume-monitor
   1438 ?        Ssl    0:00 /usr/libexec/gvfs-gphoto2-volume-monitor
   1442 ?        Ssl    0:00 /usr/libexec/gvfs-goa-volume-monitor
   1446 ?        Sl     0:00 /usr/libexec/goa-daemon
   1457 ?        Rsl    0:37 /usr/bin/gnome-shell
   1458 ?        Sl     0:00 /usr/libexec/at-spi-bus-launcher --launch-immediate
   1467 ?        S      0:00 /usr/bin/dbus-daemon --config-file=/usr/share/defau
   1476 ?        Sl     0:00 /usr/libexec/goa-identity-service
   1478 ?        Ssl    0:00 /usr/libexec/gvfs-mtp-volume-monitor
```

PID:
Process id

Durum

ps - report a snapshot of
the current processes



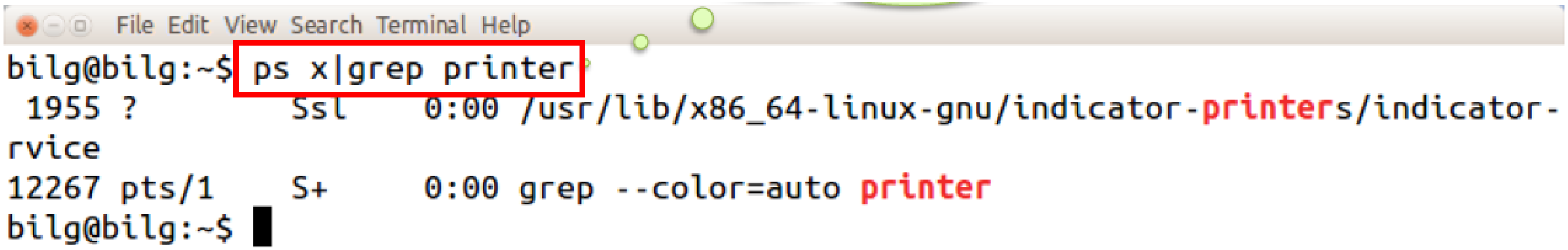
```
bilg@bilg:~/Documents/ders1a/h2$ ps x
  PID TTY      STAT   TIME COMMAND
 1599 ?        Sl     0:00 /usr/bin/gnome-keyring-daemon --daemonize --login
 1601 ?        Ss     0:00 init --user
 1661 ?        S      0:00 dbus-launch --autolaunch=9d91785545bd0650569f2c4b568c53c5 --binary
 1662 ?        Ss     0:00 //bin/dbus-daemon --fork --print-pid 5 --print-address 7 --session
 1677 ?        Ss     0:03 dbus-daemon --fork --session --address=unix:abstract=/tmp/dbus-Rw8
 1688 ?        Ss     0:00 upstart-event-bridge
 1696 ?        Ss     0:00 /usr/lib/x86_64-linux-gnu/hud/window-stack-bridge
 1699 ?        Ssl    0:15 /usr/bin/ibus-daemon --daemonize --xim
 1708 ?        Sl     0:00 /usr/lib/gvfs/gvfsd
 1714 ?        Sl     0:00 /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o big_writes
 1718 ?        Sl     0:00 /usr/lib/ibus/ibus-dconf
 1719 ?        Sl     0:06 /usr/lib/ibus/ibus-ui-gtk3
 1721 ?        Sl     0:01 /usr/lib/ibus/ibus-x11 --kill-daemon
                                        pi2-core/at-spi-bus-launcher
                                        mon --config-file=/etc/at-spi2/accessibility.conf --n
                                        pi2-core/at-spi2-registryd --use-gnome-session
                                        y-settings-daemon/unity-settings-daemon
                                        64-linux-gnu/hud/hud-service
                                        bridge --daemon --session --user --bus-name session
                                        bridge --daemon --user
                                        bridge --daemon --system --user --bus-name system
                                         --session=ubuntu
                                        y/unity-panel-service
                                        /ibus-engine-simple
                                        64-linux-gnu/bamf/bamfdaemon
                                        f/dconf-service
                                        1.0 -t -K -R
```

Process state codes
The codes used are:

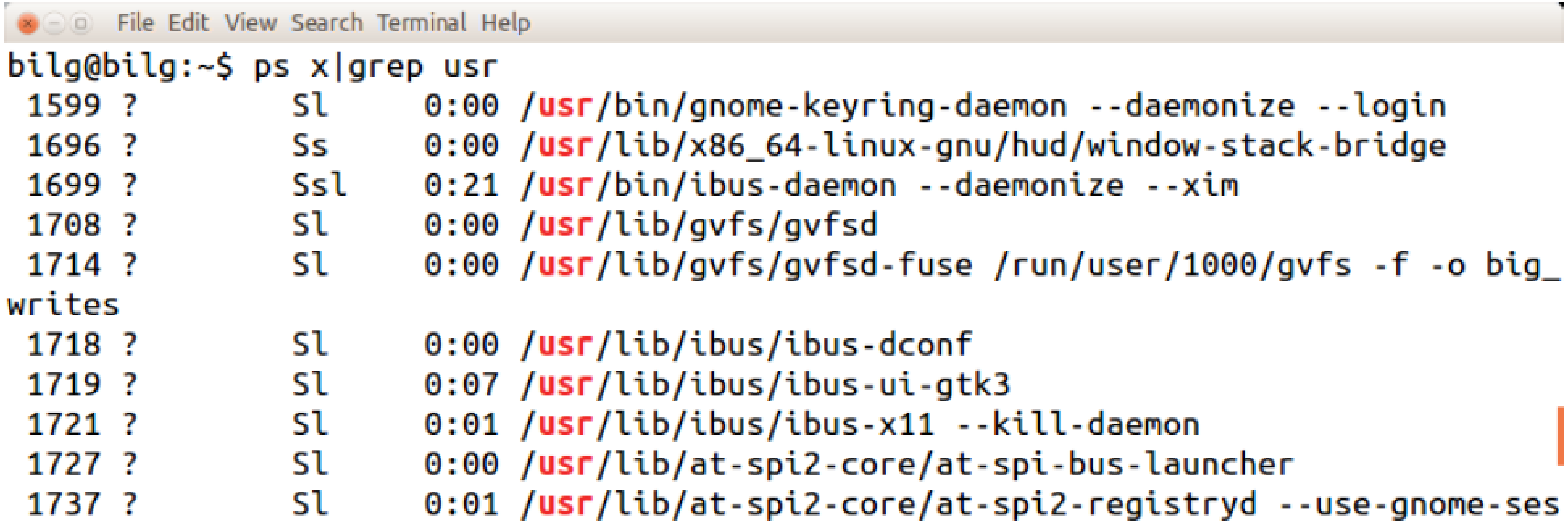
Code	Meaning
D	Uninterruptible sleep (usually IO)
R	Running or runnable (on run queue)
S	Interruptible sleep (waiting for an event to complete)
T	Stopped, either by a job control signal or because it is being traced.
W	paging (not valid since the 2.6.xx kernel)
X	dead (should never be seen)
Z	Defunct ("zombie") process, terminated but not reaped by its parent.

For BSD formats and when the stat keyword is used, additional characters may be displayed:

Code	Meaning
<	high-priority (not nice to other users)
N	low-priority (nice to other users)
L	has pages locked into memory (for real-time and custom IO)
s	is a session leader
l	is multi-threaded (using CLONE_THREAD, like NPTL pthreads do)
+	is in the foreground process group



```
bilg@bilg:~$ ps x|grep printer
 1955 ?        Ssl    0:00 /usr/lib/x86_64-linux-gnu/indicator-printers/indicator-
rvice
12267 pts/1    S+     0:00 grep --color=auto printer
bilg@bilg:~$ 
```

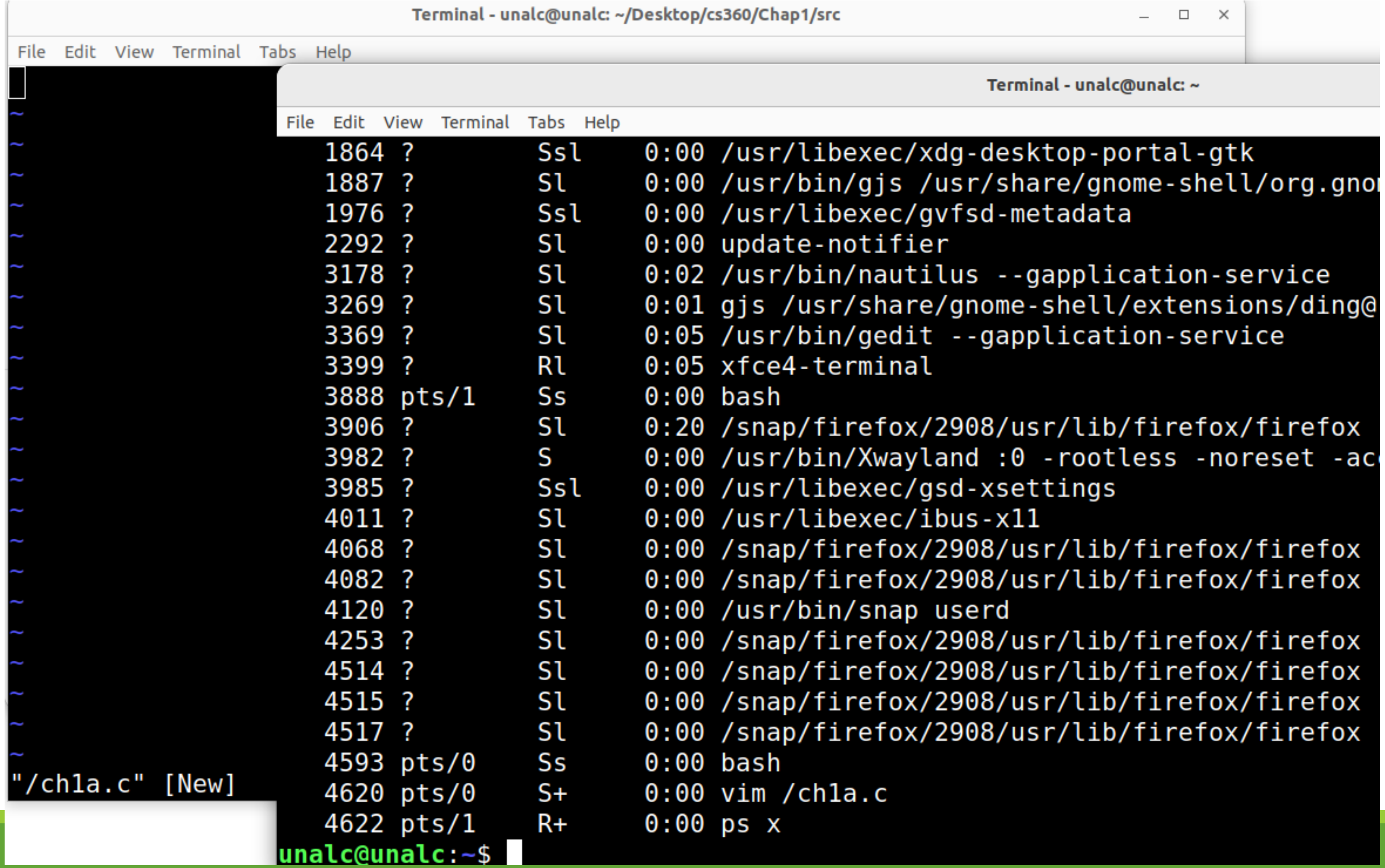


```
bilg@bilg:~$ ps x|grep usr
 1599 ?        Sl     0:00 /usr/bin/gnome-keyring-daemon --daemonize --login
 1696 ?        Ss     0:00 /usr/lib/x86_64-linux-gnu/hud/window-stack-bridge
 1699 ?        Ssl    0:21 /usr/bin/ibus-daemon --daemonize --xim
 1708 ?        Sl     0:00 /usr/lib/gvfs/gvfsd
 1714 ?        Sl     0:00 /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o big_
writes
 1718 ?        Sl     0:00 /usr/lib/ibus/ibus-dconf
 1719 ?        Sl     0:07 /usr/lib/ibus/ibus-ui-gtk3
 1721 ?        Sl     0:01 /usr/lib/ibus/ibus-x11 --kill-daemon
 1727 ?        Sl     0:00 /usr/lib/at-spi2-core/at-spi-bus-launcher
 1737 ?        Sl     0:01 /usr/lib/at-spi2-core/at-spi2-registryd --use-gnome-ses
```

Processes

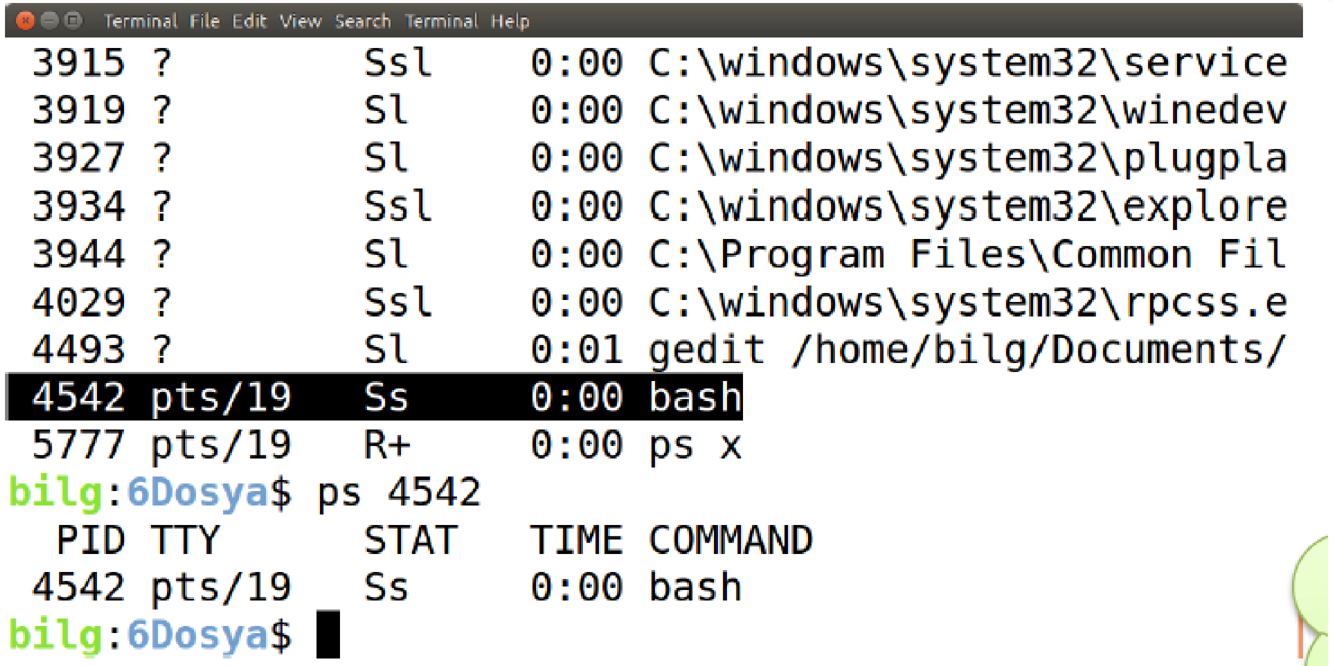
- ❑Vim editor from terminal usable and command able to receive widespread for use owner one is the editor .

- ❑Note that we can run multiple vim processes at the same time. Go to another window and type "vim $ pwd /p1.c". Now when you type " ps x " you will see the second action.

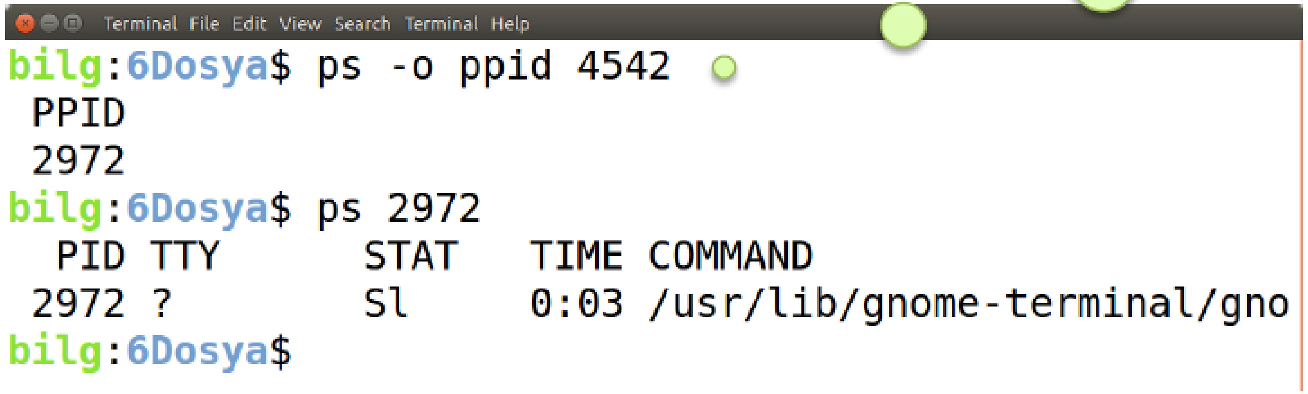

Terminal - unalc@unalc: ~/Desktop/cs360/Chap1/src
File Edit View Terminal Tabs Help
Terminal - unalc@unalc: ~
File Edit View Terminal Tabs Help

```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/ch1a.c" [New]
```

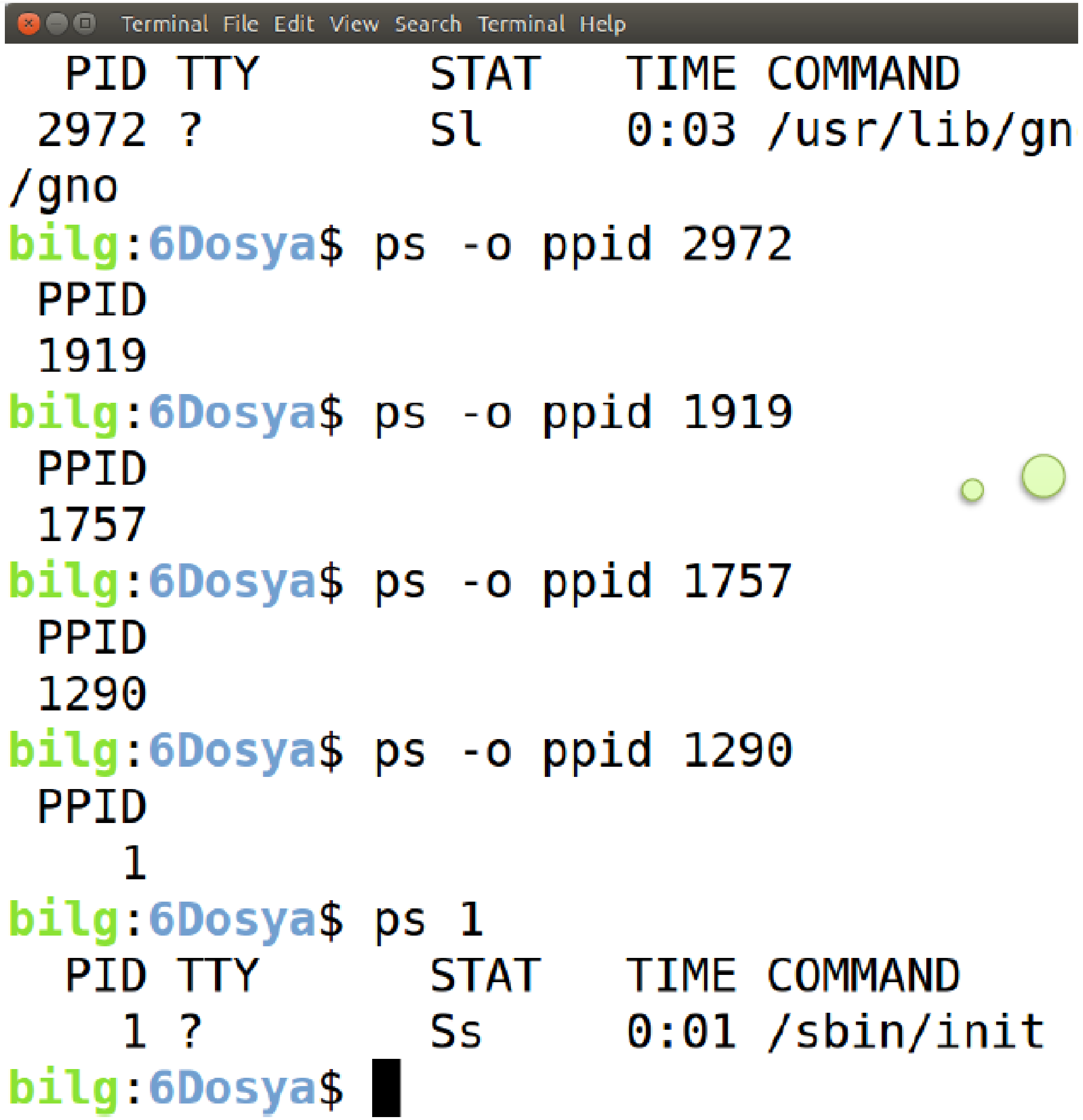
```
   1864 ?        Ssl    0:00 /usr/libexec/xdg-desktop-portal-gtk
   1887 ?        Sl     0:00 /usr/bin/gjs /usr/share/gnome-shell/org.gno
   1976 ?        Ssl    0:00 /usr/libexec/gvfsd-metadata
   2292 ?        Sl     0:00 update-notifier
   3178 ?        Sl     0:02 /usr/bin/nautilus --gapplication-service
   3269 ?        Sl     0:01 gjs /usr/share/gnome-shell/extensions/ding@
   3369 ?        Sl     0:05 /usr/bin/gedit --gapplication-service
   3399 ?        Rl     0:05 xfce4-terminal
   3888 pts/1    Ss     0:00 bash
   3906 ?        Sl     0:20 /snap/firefox/2908/usr/lib/firefox/firefox
   3982 ?        S      0:00 /usr/bin/Xwayland :0 -rootless -noreset -ac
   3985 ?        Ssl    0:00 /usr/libexec/gsd-xsettings
   4011 ?        Sl     0:00 /usr/libexec/ibus-x11
   4068 ?        Sl     0:00 /snap/firefox/2908/usr/lib/firefox/firefox
   4082 ?        Sl     0:00 /snap/firefox/2908/usr/lib/firefox/firefox
   4120 ?        Sl     0:00 /usr/bin/snap userd
   4253 ?        Sl     0:00 /snap/firefox/2908/usr/lib/firefox/firefox
   4514 ?        Sl     0:00 /snap/firefox/2908/usr/lib/firefox/firefox
   4515 ?        Sl     0:00 /snap/firefox/2908/usr/lib/firefox/firefox
   4517 ?        Sl     0:00 /snap/firefox/2908/usr/lib/firefox/firefox
   4593 pts/0    Ss     0:00 bash
   4620 pts/0    S+     0:00 vim /ch1a.c
   4622 pts/1    R+     0:00 ps x
unalc@unalc:~$ 
```



```
 3915 ?        Ssl    0:00 C:\windows\system32\service
 3919 ?        Sl     0:00 C:\windows\system32\winedev
 3927 ?        Sl     0:00 C:\windows\system32\plugpla
 3934 ?        Ssl    0:00 C:\windows\system32\explore
 3944 ?        Sl     0:00 C:\Program Files\Common Fil
 4029 ?        Ssl    0:00 C:\windows\system32\rpcss.e
 4493 ?        Sl     0:01 gedit /home/bilg/Documents/
 4542 pts/19   Ss     0:00 bash
 5777 pts/19   R+     0:00 ps x
bilg:6Dosya$ ps 4542
  PID TTY      STAT   TIME COMMAND
 4542 pts/19   Ss     0:00 bash
bilg:6Dosya$ 
```



```
bilg:6Dosya$ ps -o ppid 4542
 PPID
 2972
bilg:6Dosya$ ps 2972
  PID TTY      STAT   TIME COMMAND
 2972 ?        Sl     0:03 /usr/lib/gnome-terminal/gno
bilg:6Dosya$
```

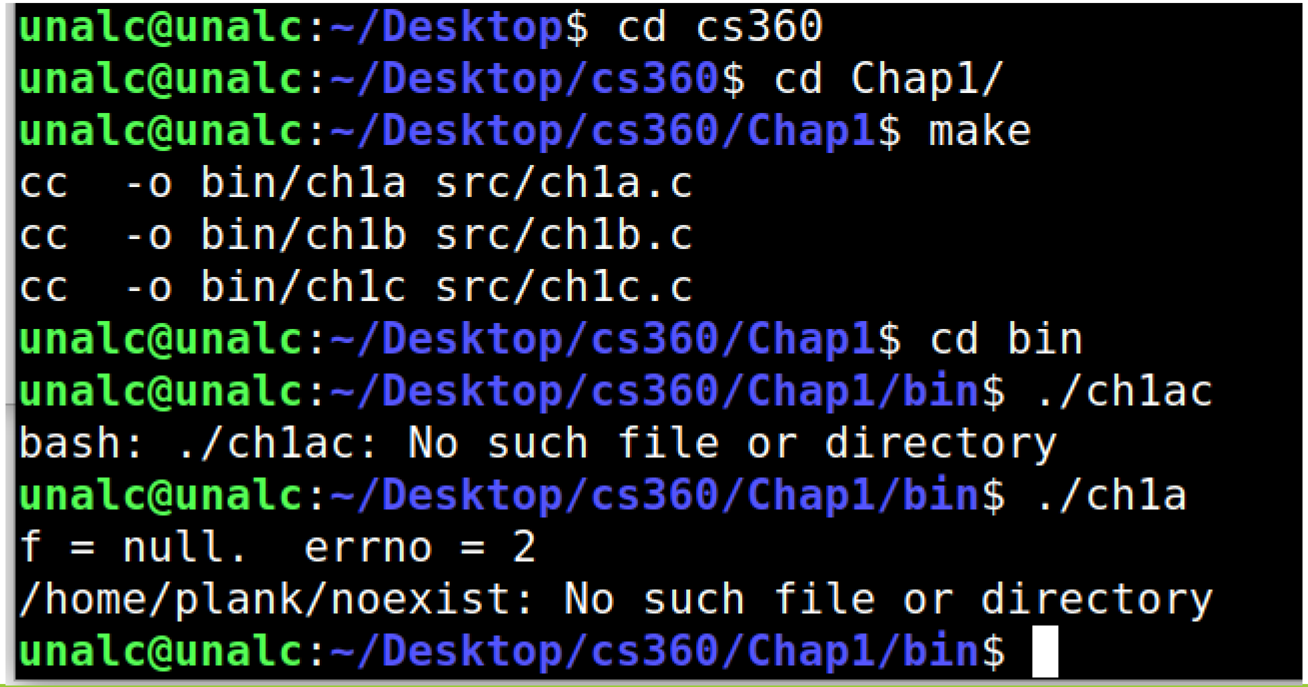


```
  PID TTY      STAT   TIME COMMAND
 2972 ?        Sl     0:03 /usr/lib/gn
/gno
bilg:6Dosya$ ps -o ppid 2972
 PPID
 1919
bilg:6Dosya$ ps -o ppid 1919
 PPID
 1757
bilg:6Dosya$ ps -o ppid 1757
 PPID
 1290
bilg:6Dosya$ ps -o ppid 1290
 PPID
    1
bilg:6Dosya$ ps 1
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
bilg:6Dosya$ 
```

Error Management

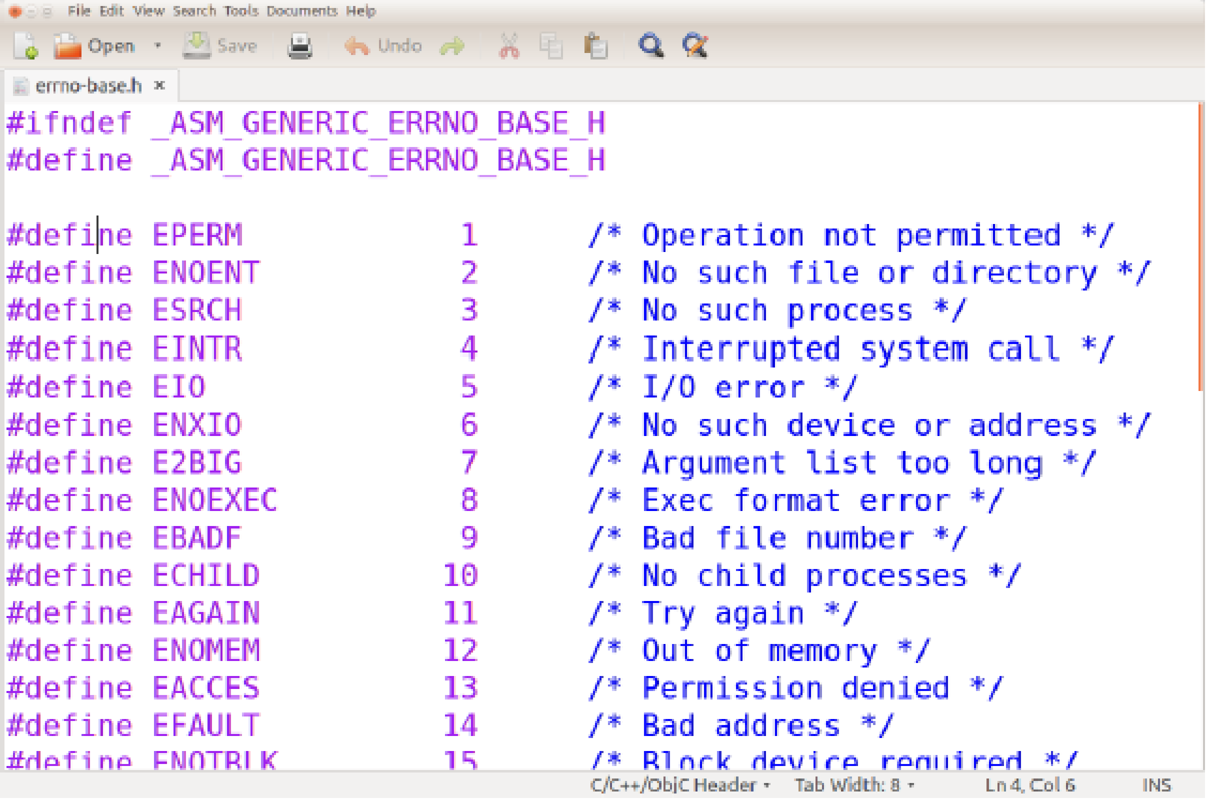
- Usually when an error occurs in a Unix system or library call, a special return value comes back, and a global variable "**errno**" is set to say what the error is.

- For example, let's say you're trying to open a file that doesn't exist:

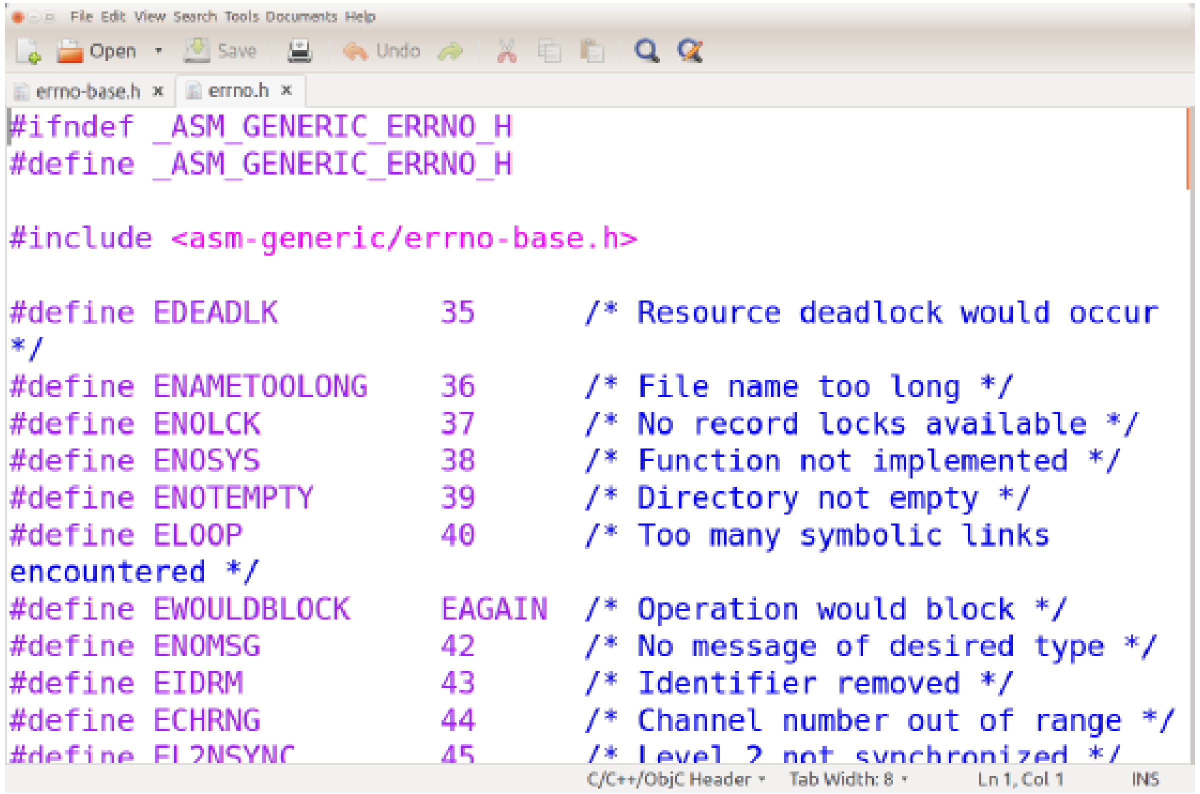


```
unalc@unalc:~/Desktop$ cd cs360
unalc@unalc:~/Desktop/cs360$ cd Chap1/
unalc@unalc:~/Desktop/cs360/Chap1$ make
cc  -o bin/ch1a src/ch1a.c
cc  -o bin/ch1b src/ch1b.c
cc  -o bin/ch1c src/ch1c.c
unalc@unalc:~/Desktop/cs360/Chap1$ cd bin
unalc@unalc:~/Desktop/cs360/Chap1/bin$ ./ch1ac
bash: ./ch1ac: No such file or directory
unalc@unalc:~/Desktop/cs360/Chap1/bin$ ./ch1a
f = null.  errno = 2
/home/plank/noexist: No such file or directory
unalc@unalc:~/Desktop/cs360/Chap1/bin$ 
```

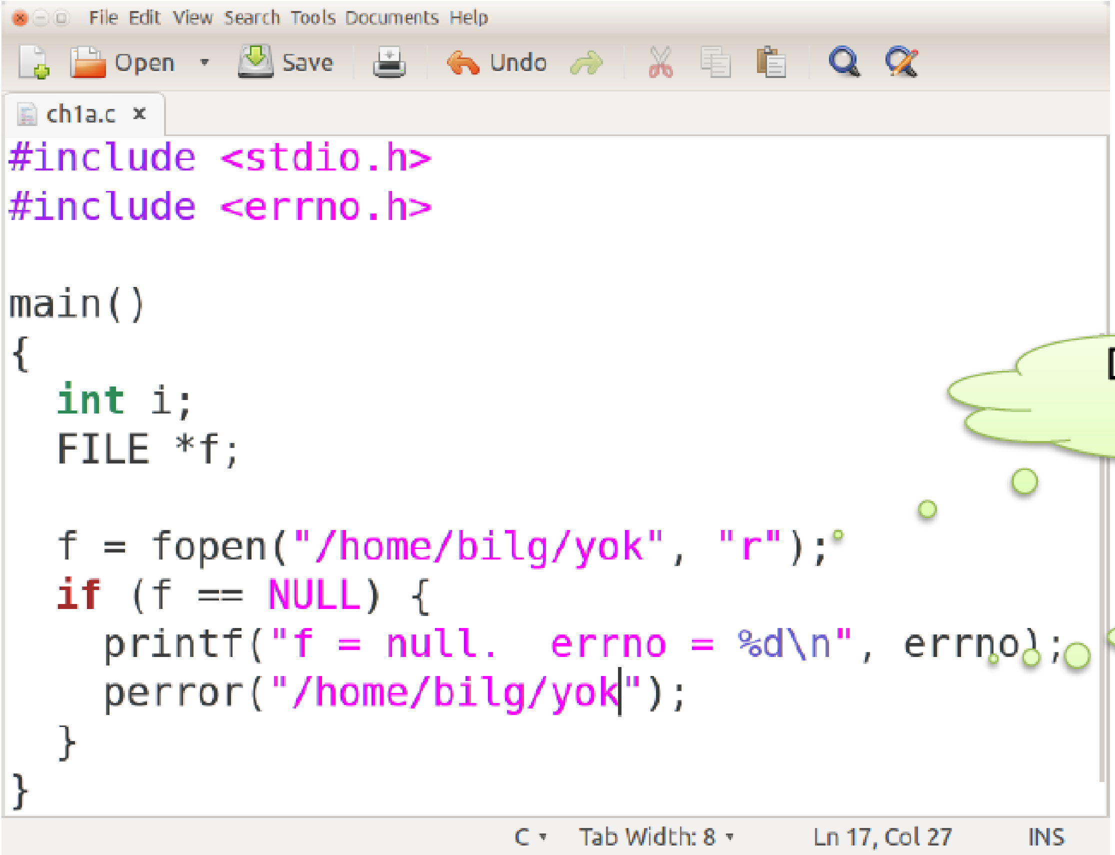
- /usr/include/asm-generic



```
#ifndef _ASM_GENERIC_ERRNO_BASE_H
#define _ASM_GENERIC_ERRNO_BASE_H

#define EPERM            1      /* Operation not permitted */
#define ENOENT           2      /* No such file or directory */
#define ESRCH            3      /* No such process */
#define EINTR            4      /* Interrupted system call */
#define EIO              5      /* I/O error */
#define ENXIO            6      /* No such device or address */
#define E2BIG            7      /* Argument list too long */
#define ENOEXEC          8      /* Exec format error */
#define EBADF            9      /* Bad file number */
#define ECHILD          10      /* No child processes */
#define EAGAIN          11      /* Try again */
#define ENOMEM          12      /* Out of memory */
#define EACCES          13      /* Permission denied */
#define EFAULT          14      /* Bad address */
#define ENOTBLK         15      /* Block device required */
```



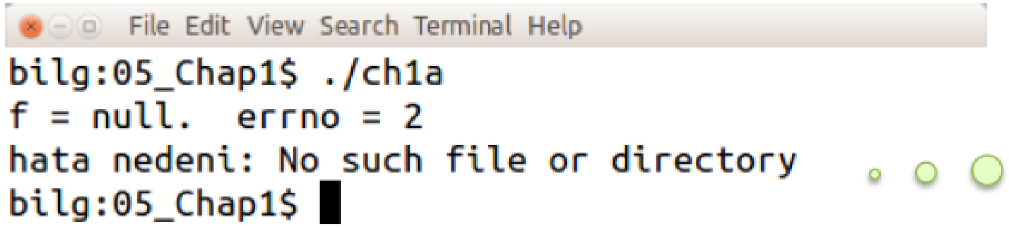
```
#ifndef _ASM_GENERIC_ERRNO_H
#define _ASM_GENERIC_ERRNO_H

#include <asm-generic/errno-base.h>

#define EDEADLK         35      /* Resource deadlock would occur
*/
#define ENAMETOOLONG    36      /* File name too long */
#define ENOLCK          37      /* No record locks available */
#define ENOSYS          38      /* Function not implemented */
#define ENOTEMPTY       39      /* Directory not empty */
#define ELOOP           40      /* Too many symbolic links
encountered */
#define EWOULDBLOCK     EAGAIN  /* Operation would block */
#define ENOMSG          42      /* No message of desired type */
#define EIDRM           43      /* Identifier removed */
#define ECHRNG          44      /* Channel number out of range */
#define EL2NSYNC        45      /* Level 2 not synchronized */
```



```c
#include <stdio.h>
#include <errno.h>

main()
{
  int i;
  FILE *f;

  f = fopen("/home/bilg/yok", "r");
  if (f == NULL) {
    printf("f = null.  errno = %d\n", errno);
    perror("/home/bilg/yok");
  }
}
```

Dosya olmadığı için
f=NULL

errno=2: olmayan dosya
hatası



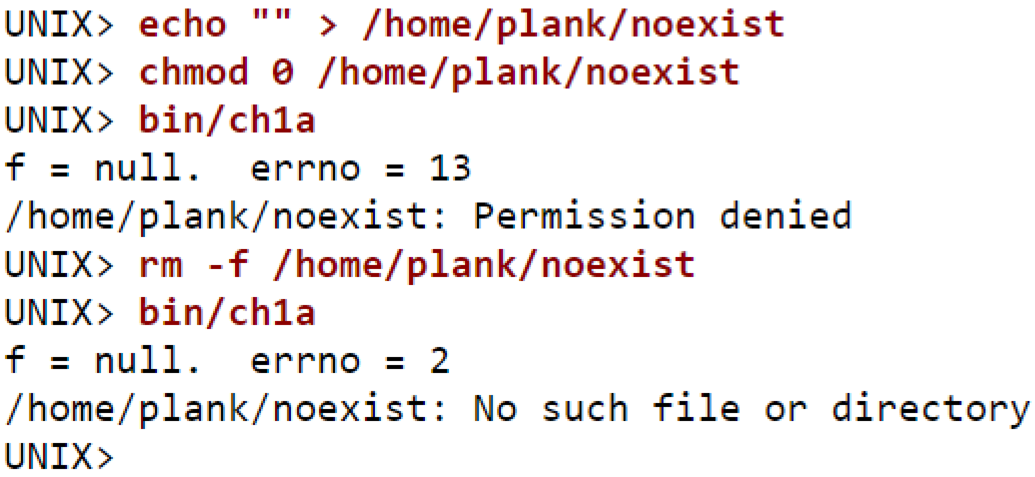
```
bilg:05_Chap1$ ./ch1a
f = null.  errno = 2
hata nedeni: No such file or directory
bilg:05_Chap1$ 
```

perror: Hata
açıklamasını yazdır

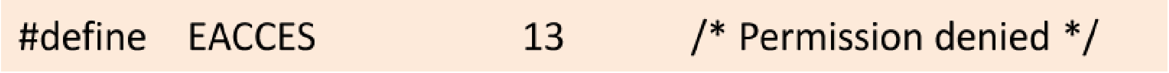
```
#define    ENOENT            2          /* No such file or directory */
```

Error handling

- ❑Let's say I created / home / plank / noexist and chmoded it so I couldn't open it for reading .
- ❑Then bin/ch1a will print a different error:

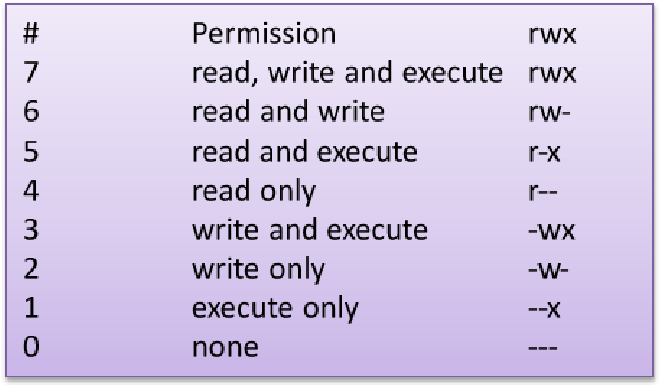


```
UNIX> echo "" > /home/plank/noexist
UNIX> chmod 0 /home/plank/noexist
UNIX> bin/ch1a
f = null.  errno = 13
/home/plank/noexist: Permission denied
UNIX> rm -f /home/plank/noexist
UNIX> bin/ch1a
f = null.  errno = 2
/home/plank/noexist: No such file or directory
UNIX>
```



```
#define    EACCES           13        /* Permission denied */
```

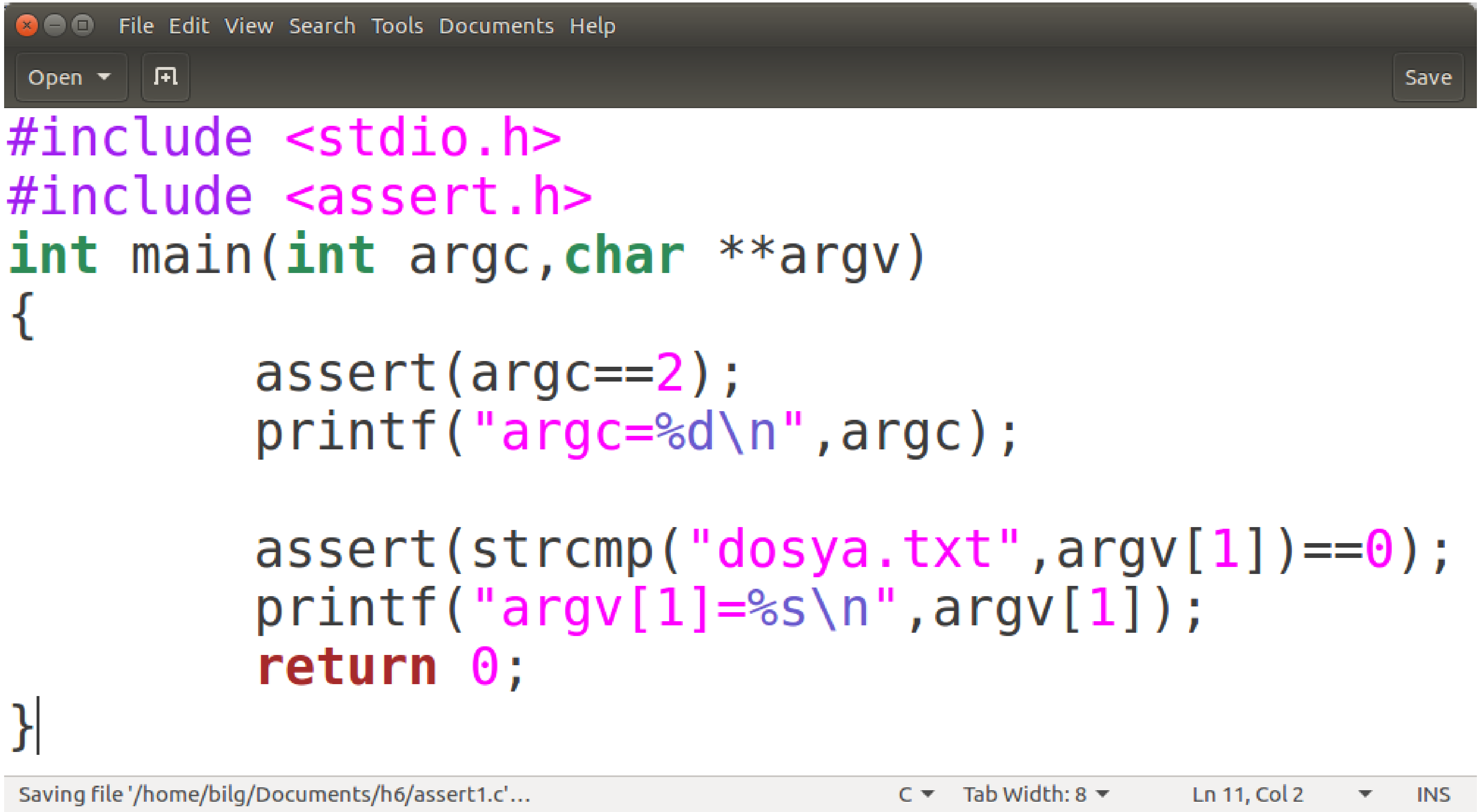
chmod - change file mode bits



#	Permission	rwx
7	read, write and execute	rwx
6	read and write	rw-
5	read and execute	r-x
4	read only	r--
3	write and execute	-wx
2	write only	-w-
1	execute only	--x
0	none	---

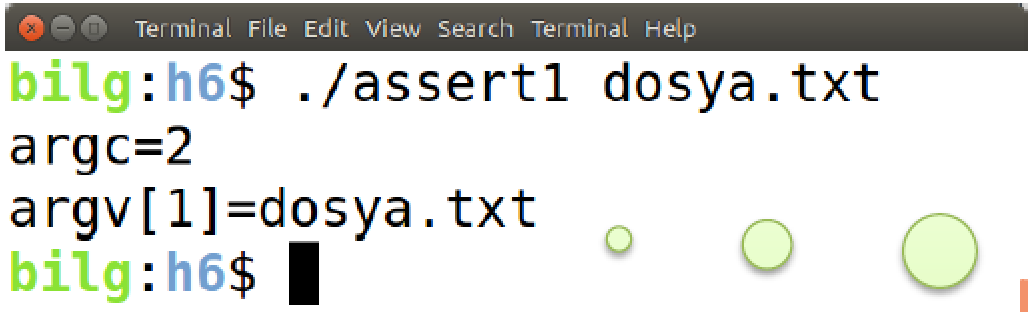
Assert

If the defined condition is false, the program flow is stopped.

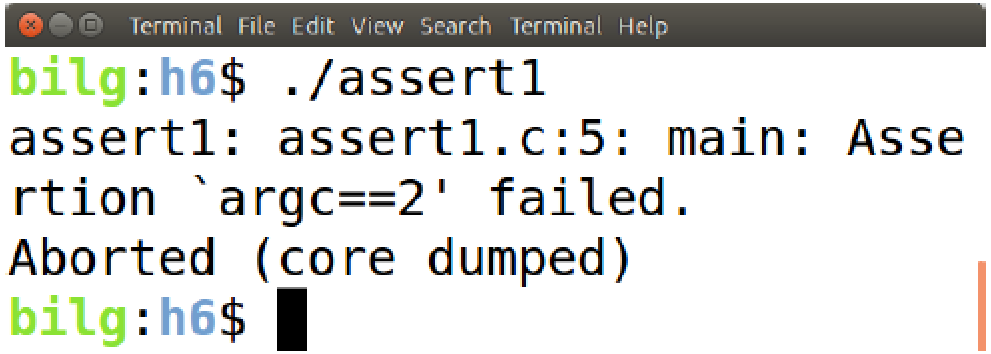


```
#include <stdio.h>
#include <assert.h>
int main(int argc,char **argv)
{
	assert(argc==2);
	printf("argc=%d\n",argc);

	assert(strcmp("dosya.txt",argv[1])==0);
	printf("argv[1]=%s\n",argv[1]);
	return 0;
}
```

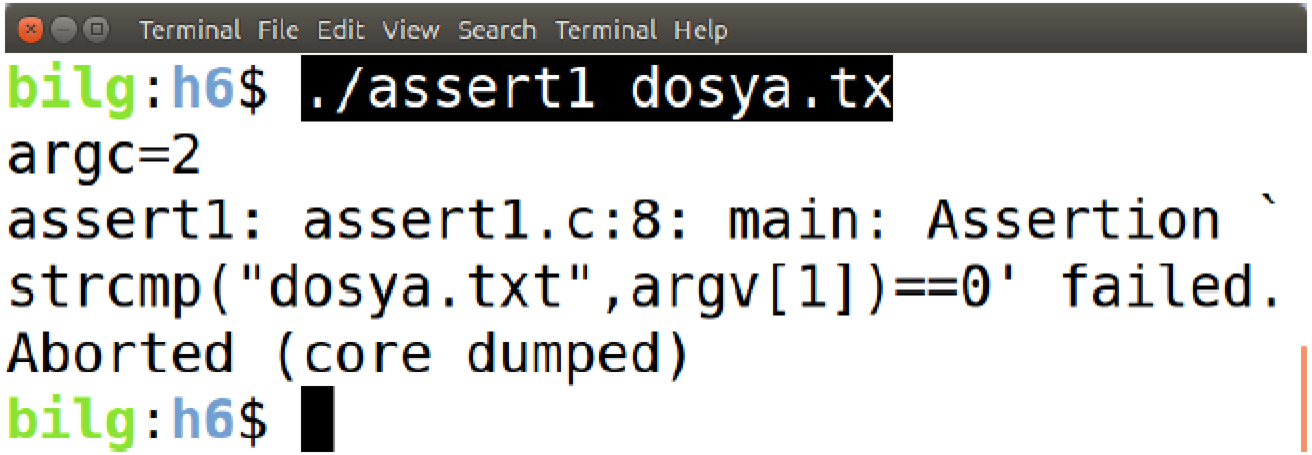
Saving file '/home/bilg/Documents/h6/assert1.c'...



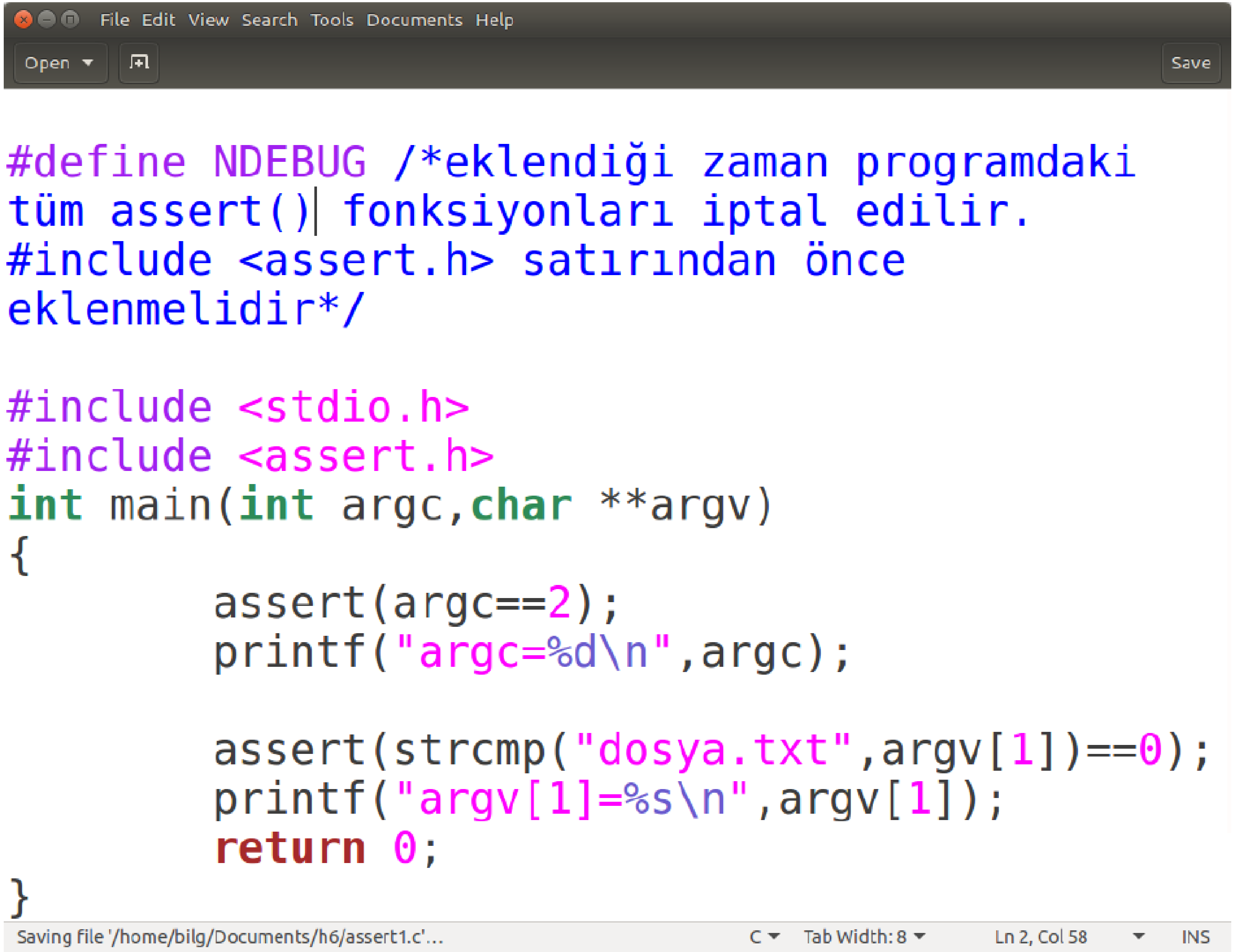
```
bilg:h6$ ./assert1 dosya.txt
argc=2
argv[1]=dosya.txt
bilg:h6$ 
```



```
bilg:h6$ ./assert1
assert1: assert1.c:5: main: Asse
rtion `argc==2' failed.
Aborted (core dumped)
bilg:h6$ 
```



```
bilg:h6$ ./assert1 dosya.tx
argc=2
assert1: assert1.c:8: main: Assertion `
strcmp("dosya.txt",argv[1])==0' failed.
Aborted (core dumped)
bilg:h6$ 
```



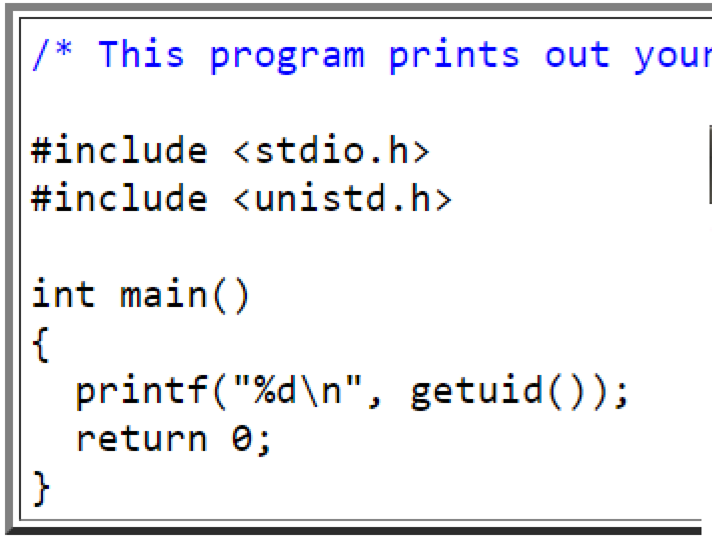
```c
#define NDEBUG /*eklendiği zaman programdaki
tüm assert() fonksiyonları iptal edilir.
#include <assert.h> satırından önce
eklenmelidir*/

#include <stdio.h>
#include <assert.h>
int main(int argc,char **argv)
{
        assert(argc==2);
        printf("argc=%d\n",argc);

        assert(strcmp("dosya.txt",argv[1])==0);
        printf("argv[1]=%s\n",argv[1]);
        return 0;
}
```

Saving file '/home/bilg/Documents/h6/assert1.c'...

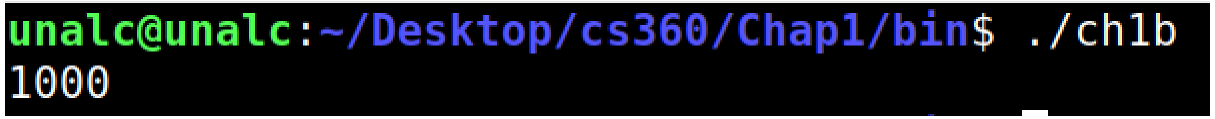
C   Tab Width: 8   Ln 2, Col 58   INS

getuid(), getgid(),getpid(),getppid()

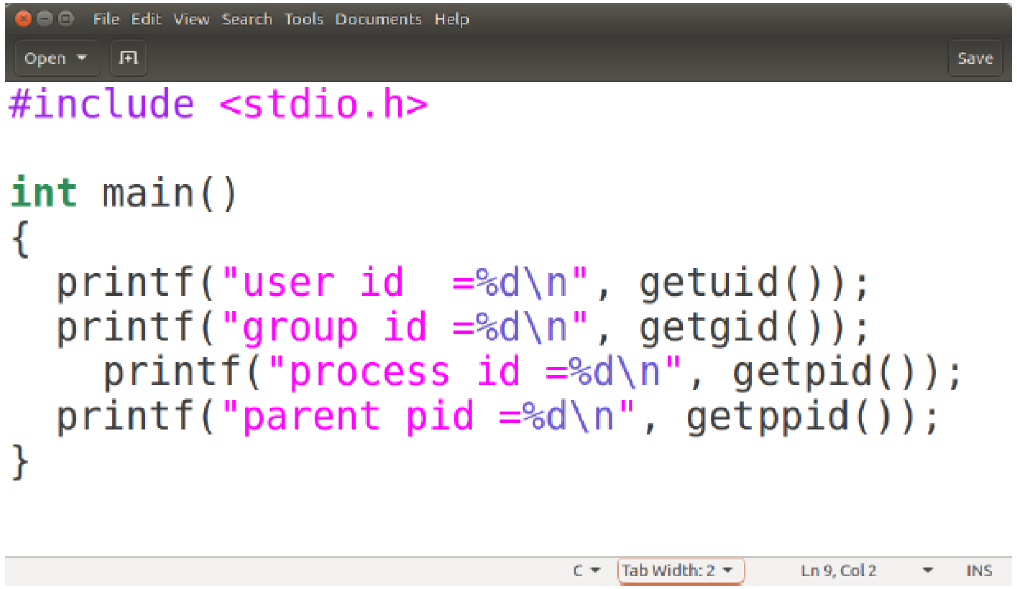
- ❑User ID: The number given to each user by the system administrator.



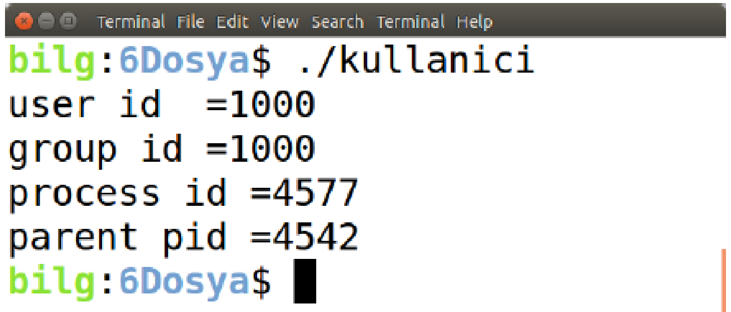
```
/* This program prints out your user id.

#include <stdio.h>
#include <unistd.h>

int main()
{
  printf("%d\n", getuid());
  return 0;
}
```



```
unalc@unalc:~/Desktop/cs360/Chap1/bin$ ./ch1b
1000
```



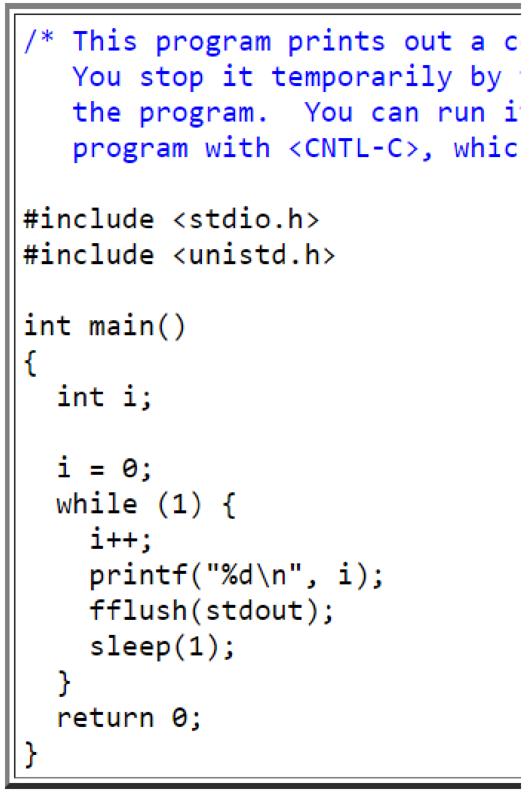
```
#include <stdio.h>

int main()
{
  printf("user id  =%d\n", getuid());
  printf("group id =%d\n", getgid());
    printf("process id =%d\n", getpid());
  printf("parent pid =%d\n", getppid());
}
```



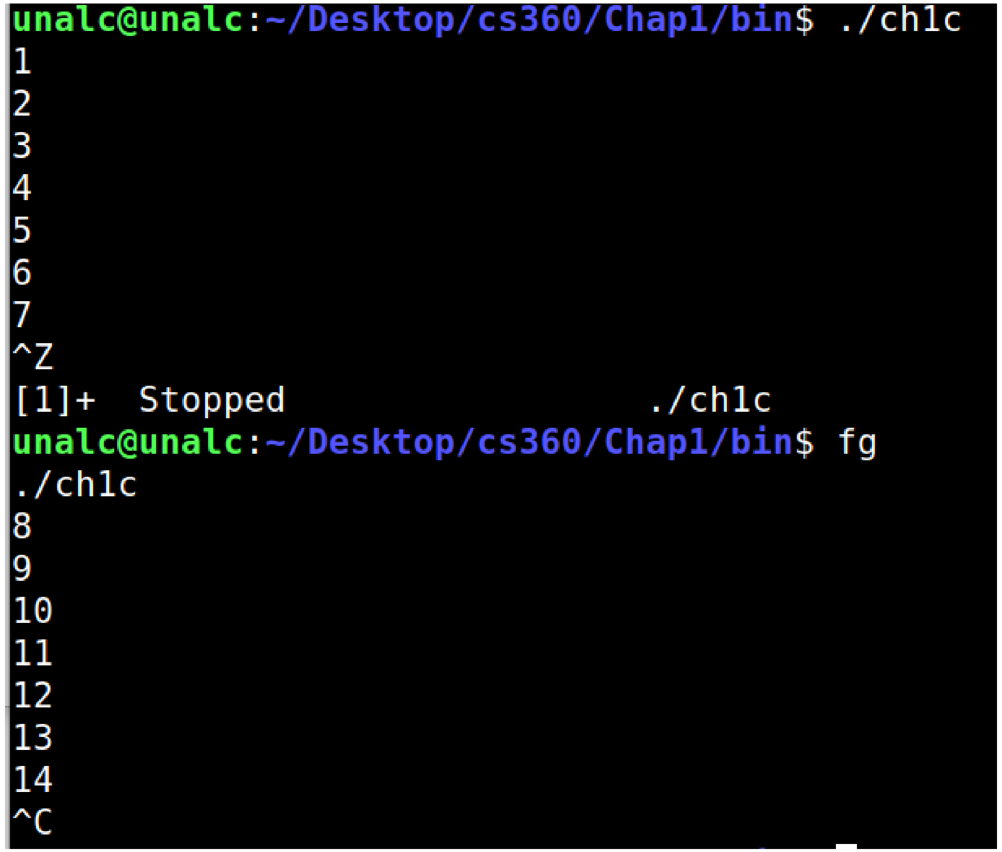
```
bilg:6Dosya$ ./kullanici
user id  =1000
group id =1000
process id =4577
parent pid =4542
bilg:6Dosya$
```

Signals

- ❑Signal: An interrupt in the program
- ❑Signal Processor (Signal Handler): The mechanism by which the program can deal with signals.
- ❑This program prints a counter that increases every second. You temporarily stop the SIGSTOP signal by typing <CNTL-Z>, which sends it to the program.
- ❑You can then terminate the program with <CNTL-C>, which sends the SIGINT signal. fg



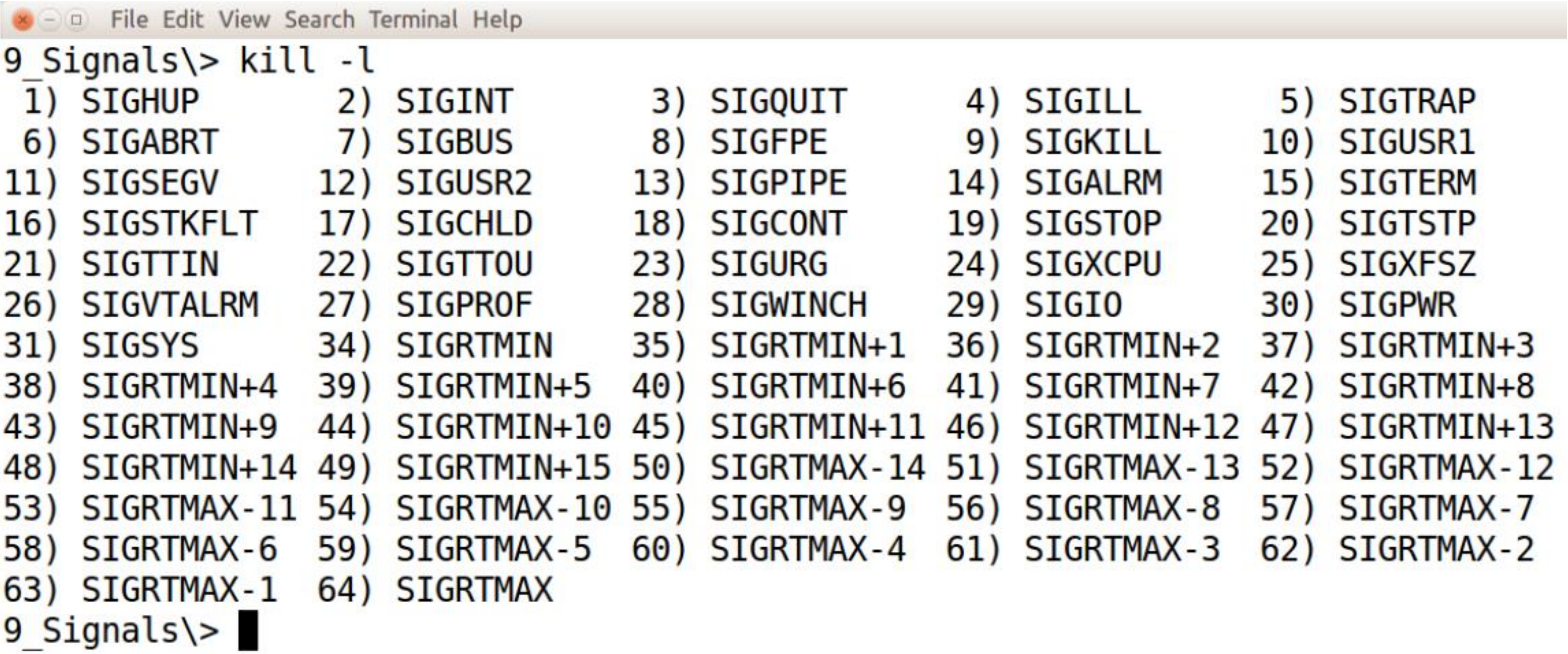
```
/* This program prints out a c
   You stop it temporarily by 
   the program.  You can run i
   program with <CNTL-C>, whic

#include <stdio.h>
#include <unistd.h>

int main()
{
  int i;

  i = 0;
  while (1) {
    i++;
    printf("%d\n", i);
    fflush(stdout);
    sleep(1);
  }
  return 0;
}
```



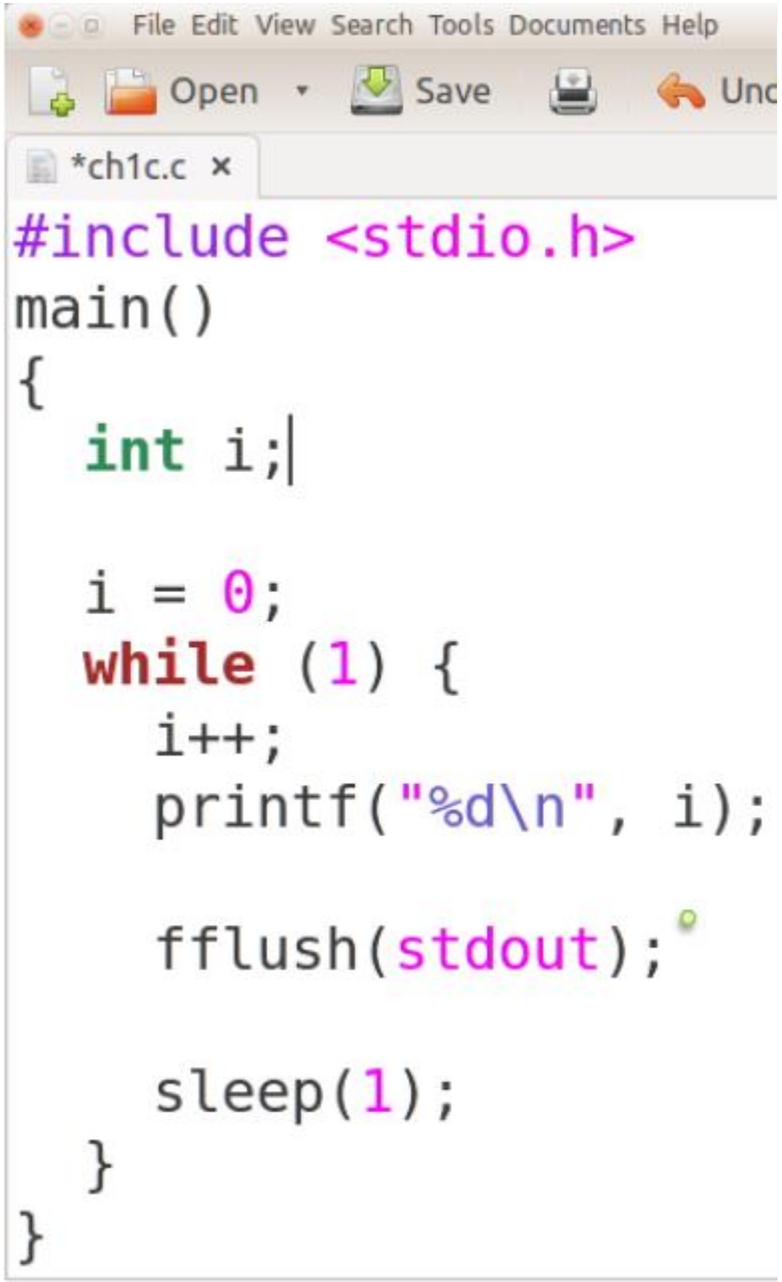
```
unalc@unalc:~/Desktop/cs360/Chap1/bin$ ./ch1c
1
2
3
4
5
6
7
^Z
[1]+  Stopped                 ./ch1c
unalc@unalc:~/Desktop/cs360/Chap1/bin$ fg
./ch1c
8
9
10
11
12
13
14
^C
```

Signals

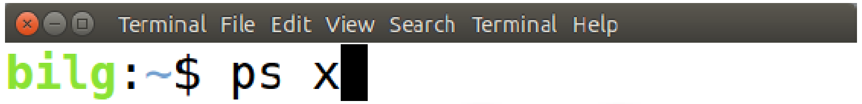
- This program implements a counter that increments itself every second.
- Let it run for a few seconds and then type < CNTL-Z >. This sends a "STOP" signal to the program which stops it. Now you'll go back to your shell. If you type " ps ", you'll see something like this:
- “T” means the process is not running – it has been stopped.
- To start, you can type “ fg ” which will send the “START” signal.
- Now, while it's running, type < CNTL-C > to terminate the program -- this sends it the "INT" signal, which ensures it. Segmentation errors are also signals.



```
9_Signals\> kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
9_Signals\> 
```

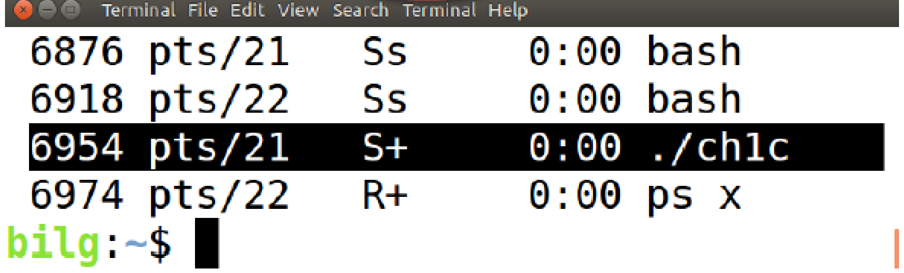


```c
#include <stdio.h>
main()
{
  int i;

  i = 0;
  while (1) {
    i++;
    printf("%d\n", i);

    fflush(stdout);

    sleep(1);
  }
}
```

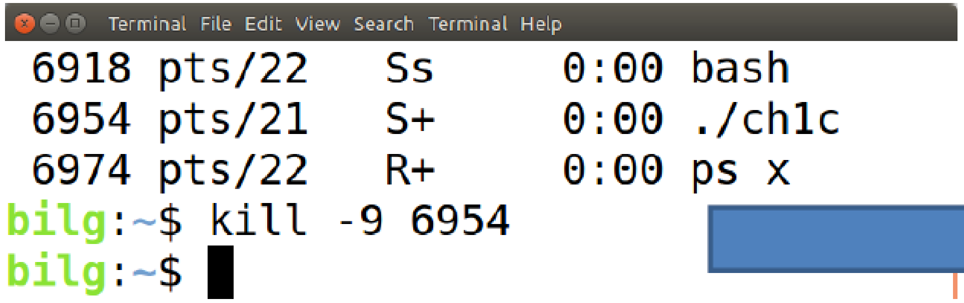


```
bilg:~$ ps x
```

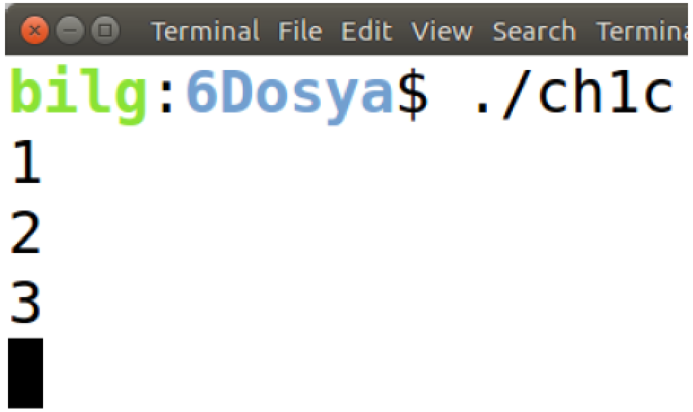
./ch1c programını çalıştırıdıktan sonra başka bir terminal ekranı açıp ps x yazarsakaşağıdaki gibi ./ch1c programını görürüz. Bu prosese kill ile sinyal gönderebiliriz.



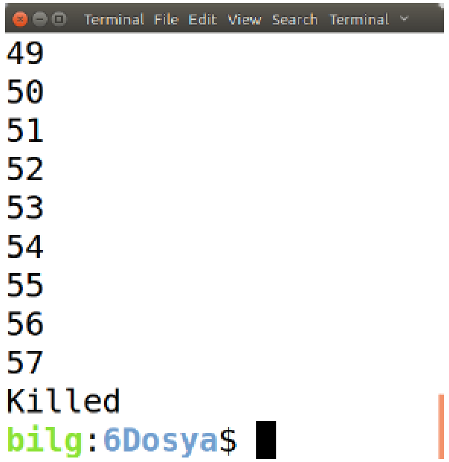
```
 6876 pts/21    Ss     0:00 bash
 6918 pts/22    Ss     0:00 bash
 6954 pts/21    S+     0:00 ./ch1c
 6974 pts/22    R+     0:00 ps x
bilg:~$ 
```



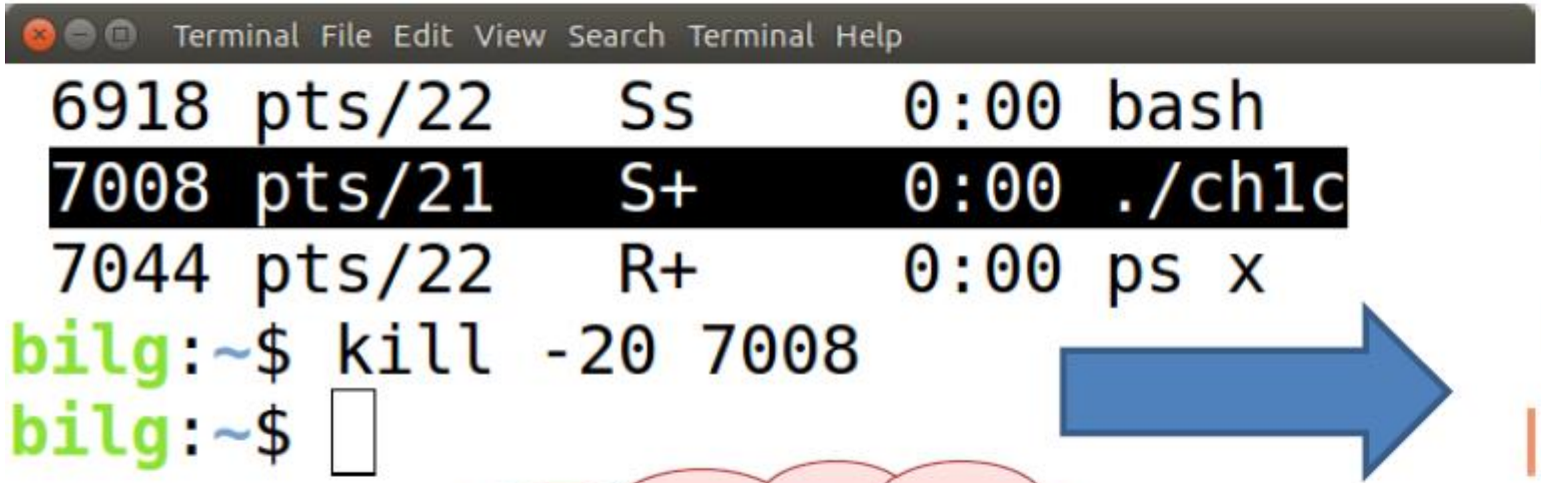
```
 6918 pts/22    Ss     0:00 bash
 6954 pts/21    S+     0:00 ./ch1c
 6974 pts/22    R+     0:00 ps x
bilg:~$ kill -9 6954
bilg:~$ 
```



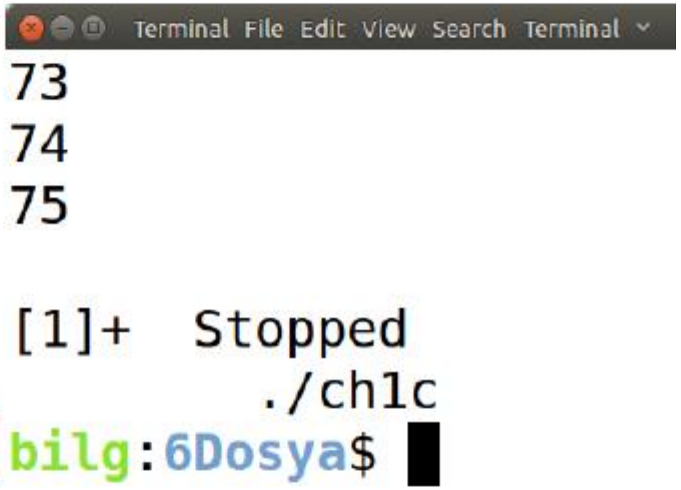
```
bilg:6Dosya$ ./ch1c
1
2
3
```



```
49
50
51
52
53
54
55
56
57
Killed
bilg:6Dosya$ 
```

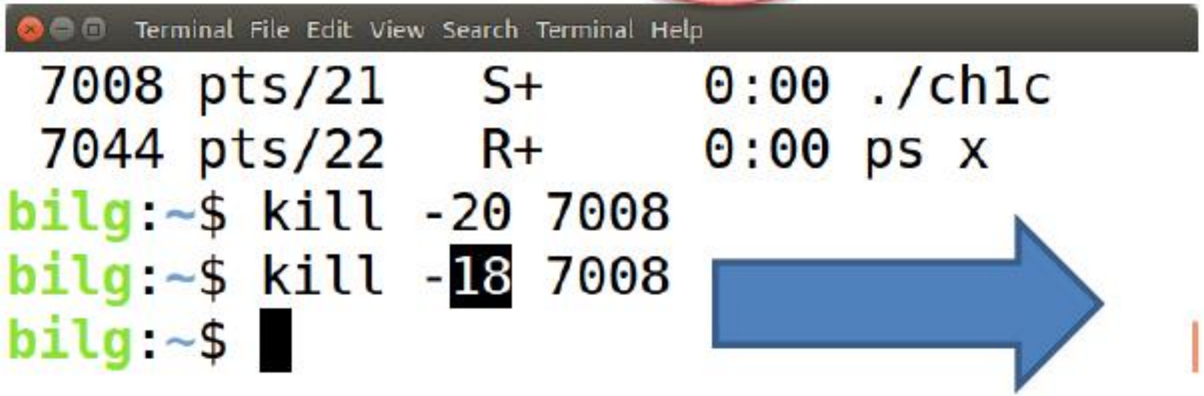


```
 6918 pts/22    Ss      0:00 bash
 7008 pts/21    S+      0:00 ./ch1c
 7044 pts/22    R+      0:00 ps x
bilg:~$ kill -20 7008
bilg:~$ 
```

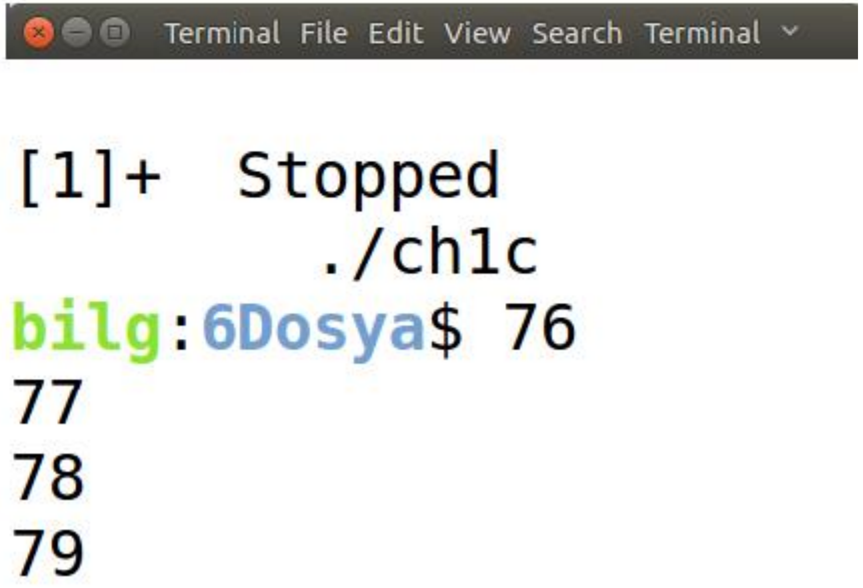


```
73
74
75

[1]+  Stopped
          ./ch1c
bilg:6Dosya$ 
```

20 ile SIGSTOP gönderdik.
18 ile SIGCONT gönderdik.

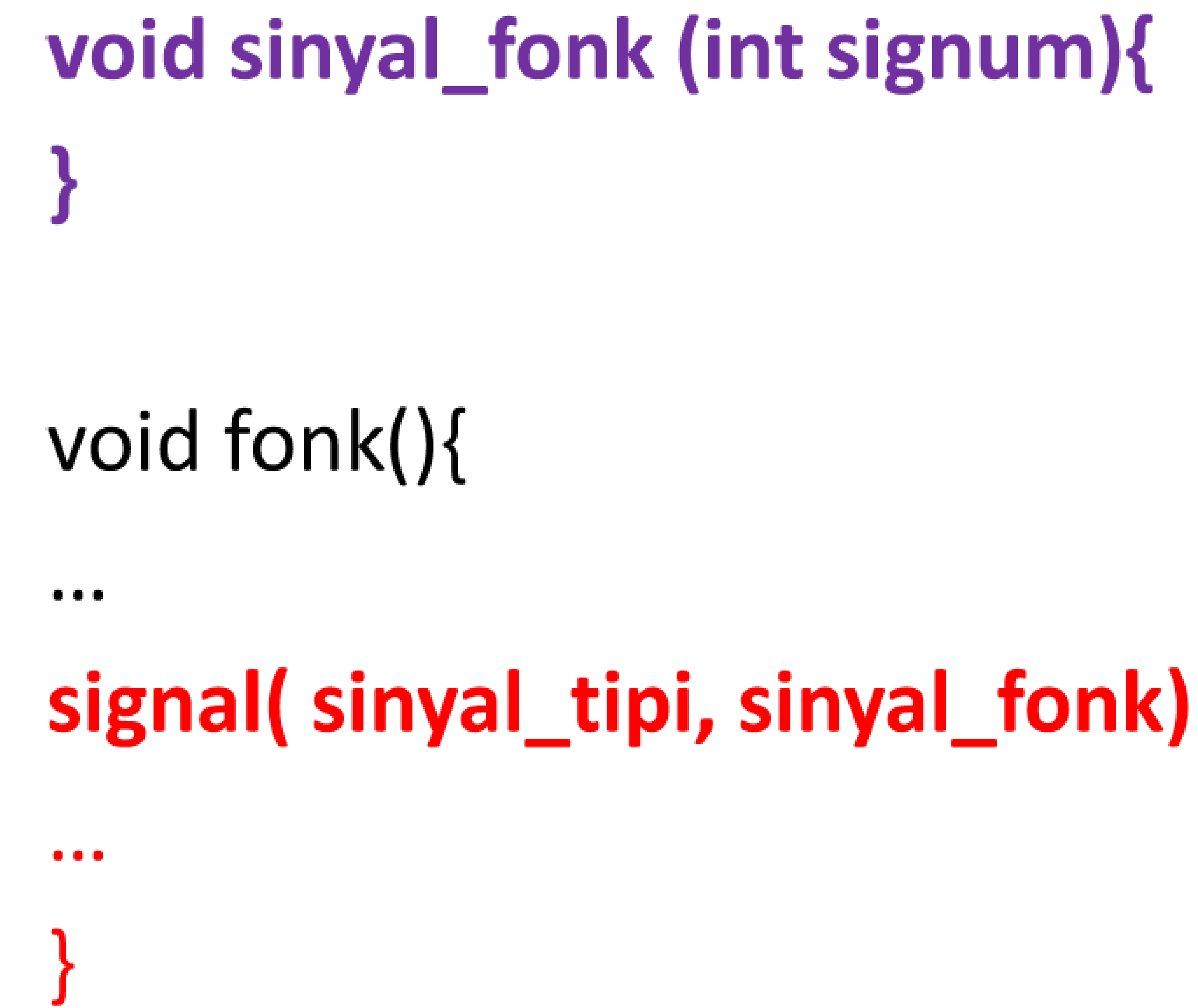


```
 7008 pts/21    S+      0:00 ./ch1c
 7044 pts/22    R+      0:00 ps x
bilg:~$ kill -20 7008
bilg:~$ kill -18 7008
bilg:~$ 
```

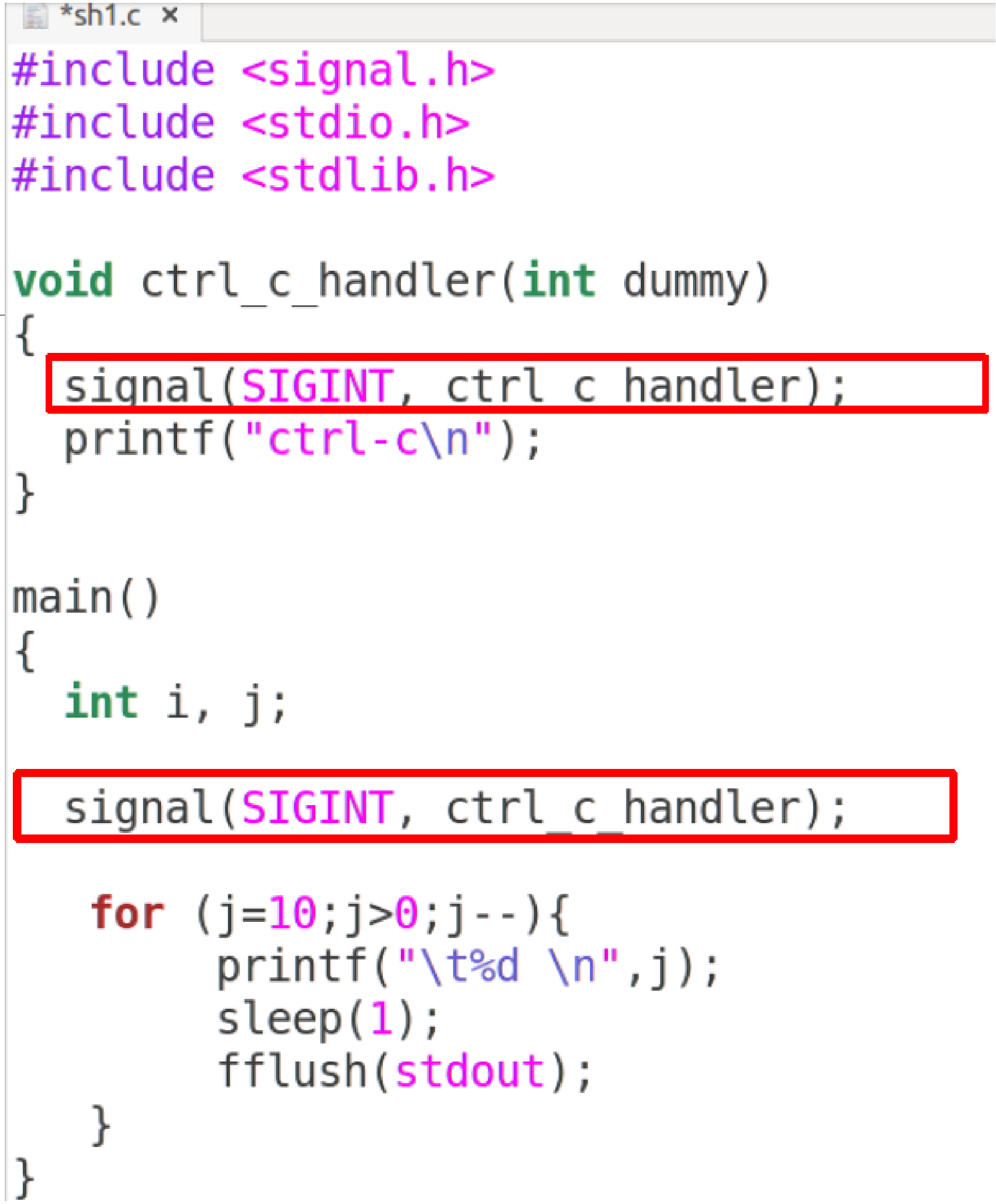


```
[1]+  Stopped
          ./ch1c
bilg:6Dosya$ 76
77
78
79
```

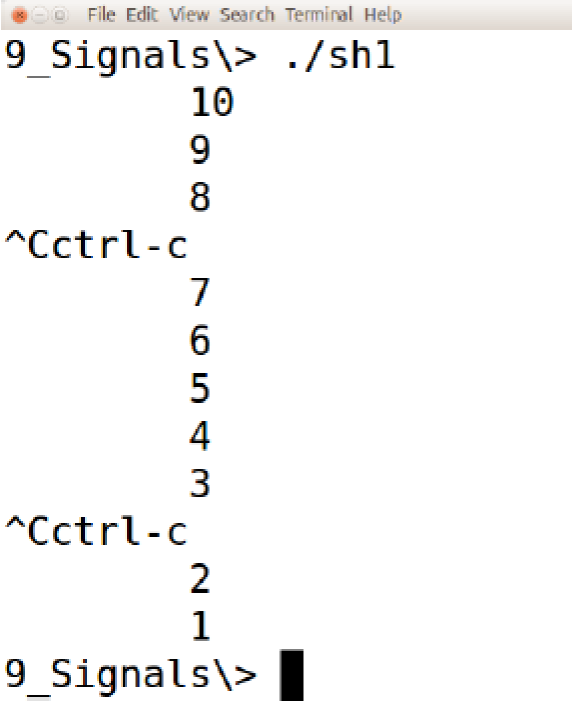
To Define Signal Function Prototype



```
void sinyal_fonk (int signum){
}


void fonk(){
…
signal( sinyal_tipi, sinyal_fonk)
…
}
```

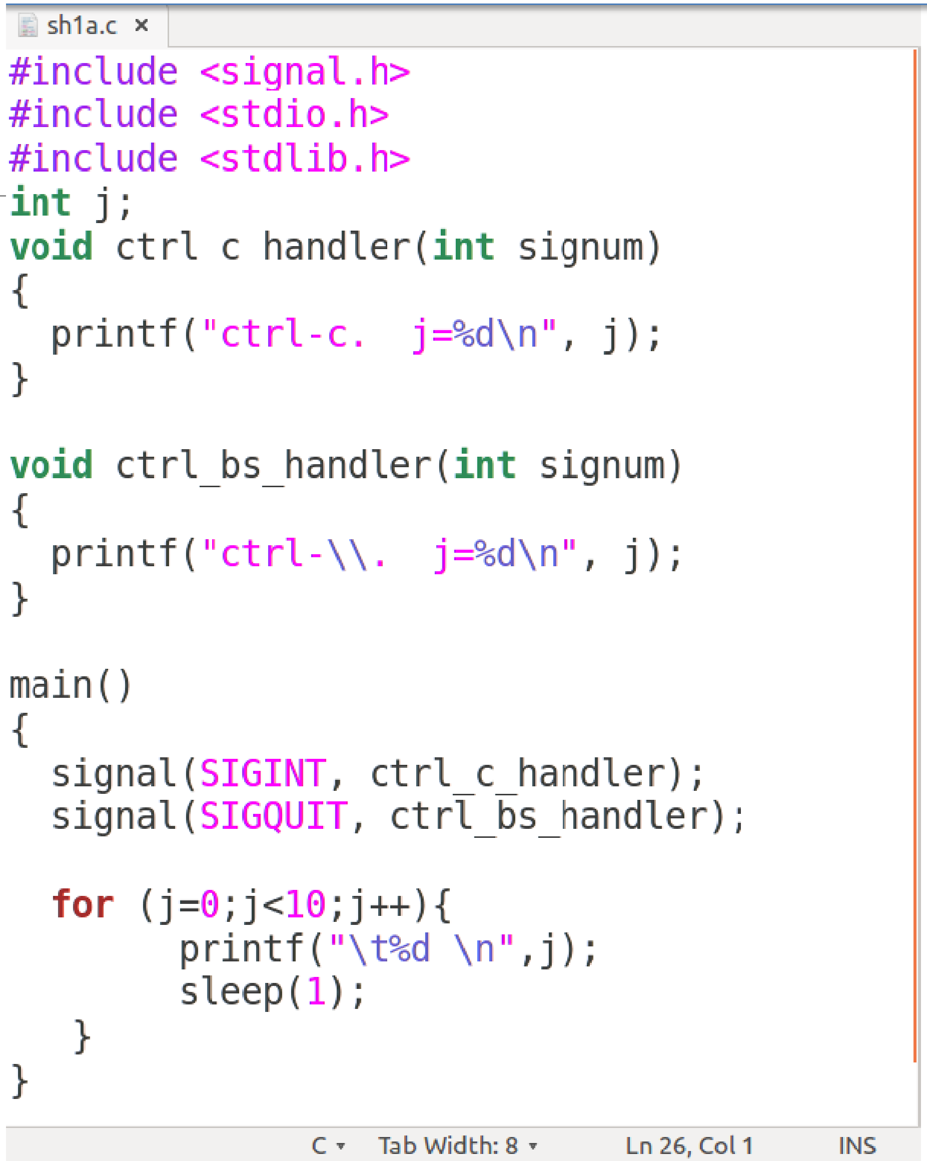


```c
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>

void ctrl_c_handler(int dummy)
{
  signal(SIGINT, ctrl_c_handler);
  printf("ctrl-c\n");
}

main()
{
  int i, j;

  signal(SIGINT, ctrl_c_handler);

   for (j=10;j>0;j--){
        printf("\t%d \n",j);
        sleep(1);
        fflush(stdout);
   }
}
```



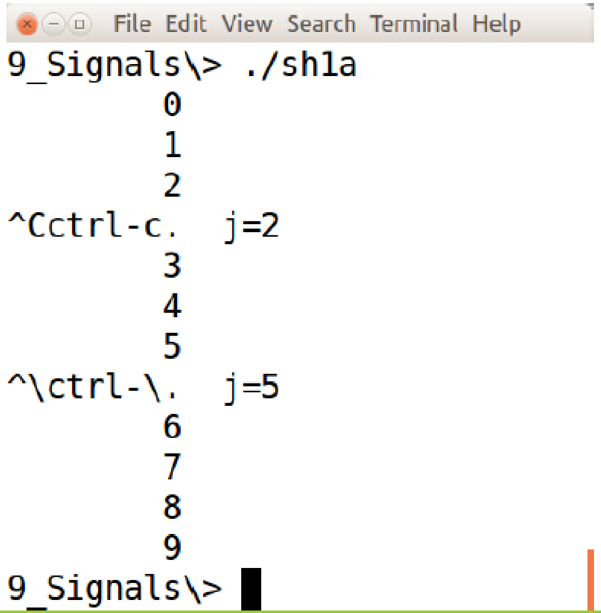
```
9_Signals\> ./sh1
        10
        9
        8
^Cctrl-c
        7
        6
        5
        4
        3
^Cctrl-c
        2
        1
9_Signals\> 
```

Multi Signal Application

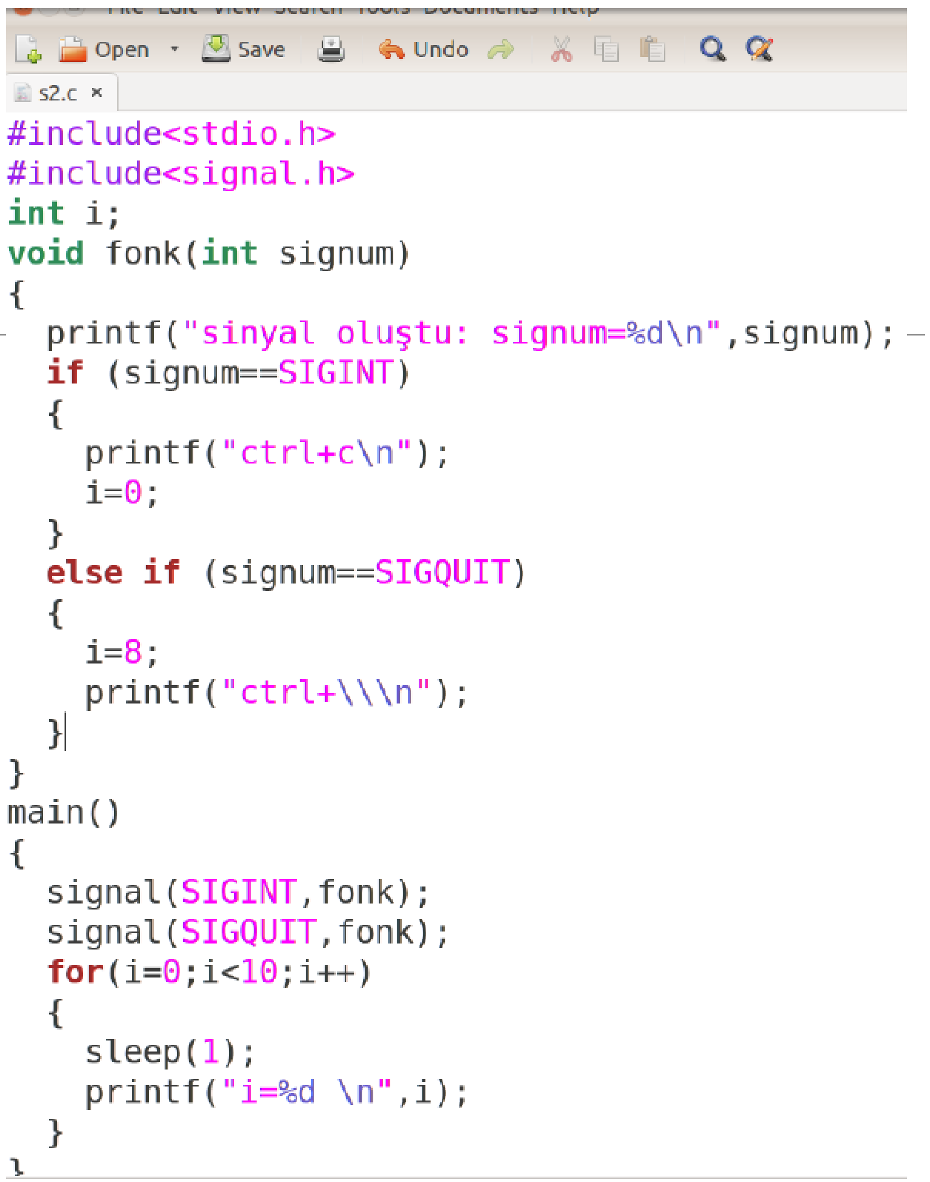


```c
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
int j;
void ctrl_c_handler(int signum)
{
  printf("ctrl-c.  j=%d\n", j);
}

void ctrl_bs_handler(int signum)
{
  printf("ctrl-\\.  j=%d\n", j);
}

main()
{
  signal(SIGINT, ctrl_c_handler);
  signal(SIGQUIT, ctrl_bs_handler);

  for (j=0;j<10;j++){
        printf("\t%d \n",j);
        sleep(1);
   }
}
```

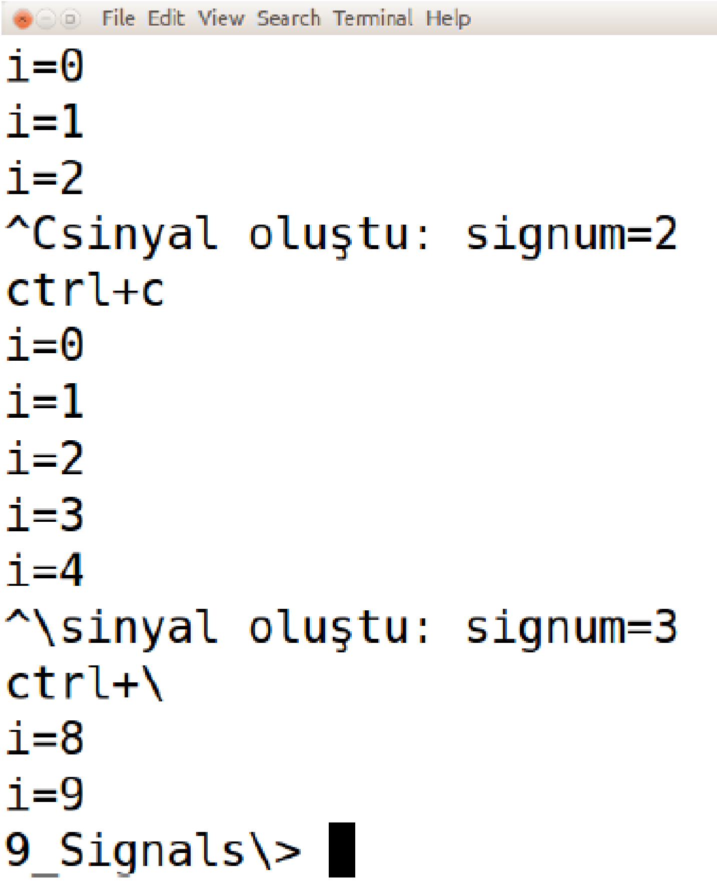
- Örnekte SIGINT yanında SIGQUIT sinyali içinde bir bir interrupt handler yazılmıştır.
- Ctrl+\ ile SIGQUIT üretiliyor.



```
9_Signals\> ./sh1a
        0
        1
        2
^Cctrl-c.  j=2
        3
        4
        5
^\ctrl-\.  j=5
        6
        7
        8
        9
9_Signals\> 
```

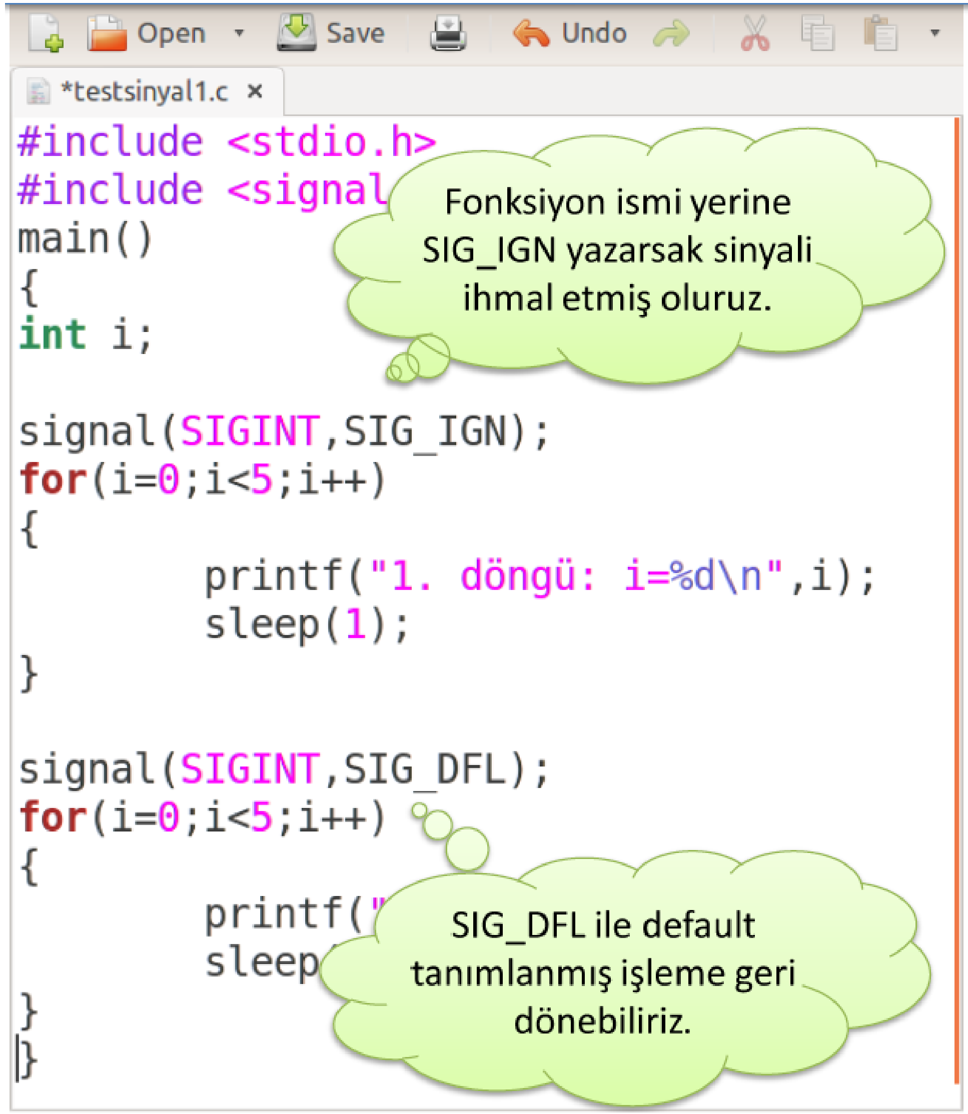


```c
#include<stdio.h>
#include<signal.h>
int i;
void fonk(int signum)
{
  printf("sinyal oluştu: signum=%d\n",signum);
  if (signum==SIGINT)
  {
    printf("ctrl+c\n");
    i=0;
  }
  else if (signum==SIGQUIT)
  {
    i=8;
    printf("ctrl+\\\n");
  }
}
main()
{
  signal(SIGINT,fonk);
  signal(SIGQUIT,fonk);
  for(i=0;i<10;i++)
  {
    sleep(1);
    printf("i=%d \n",i);
  }
}
```



```
i=0
i=1
i=2
^Csinyal oluştu: signum=2
ctrl+c
i=0
i=1
i=2
i=3
i=4
^\sinyal oluştu: signum=3
ctrl+\
i=8
i=9
9_Signals\>
```

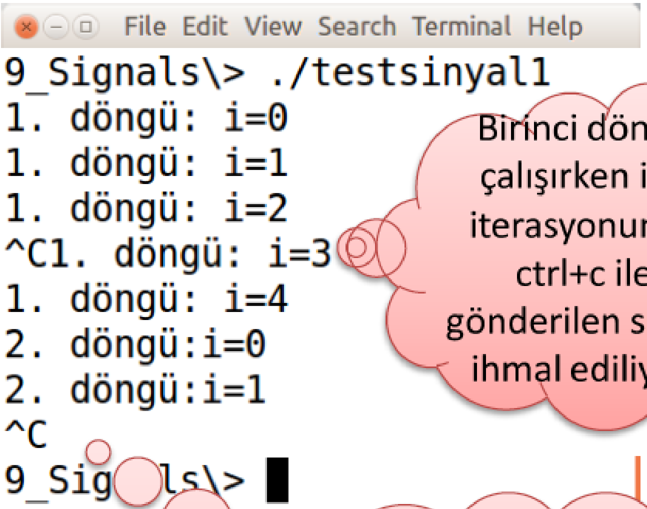
Ignoring the signal.



```
#include <stdio.h>
#include <signal
main()
{
int i;

signal(SIGINT,SIG_IGN);
for(i=0;i<5;i++)
{
        printf("1. döngü: i=%d\n",i);
        sleep(1);
}

signal(SIGINT,SIG_DFL);
for(i=0;i<5;i++)
{
        printf("
        sleep
}
}
```

Fonksiyon ismi yerine SIG_IGN yazarsak sinyali ihmal etmiş oluruz.

SIG_DFL ile default tanımlanmış işleme geri dönebiliriz.

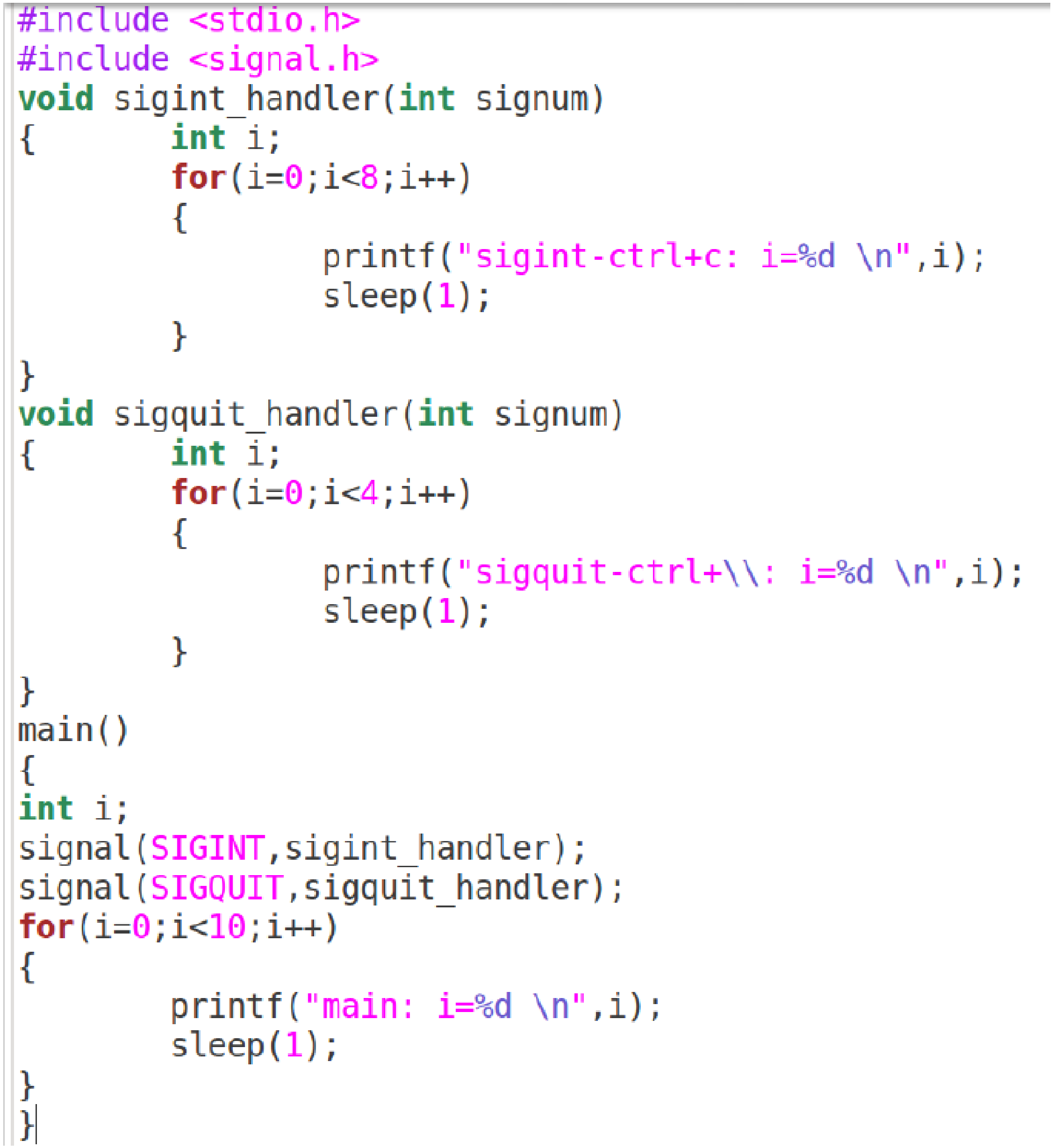


```
9_Signals\> ./testsinyal1
1. döngü: i=0
1. döngü: i=1
1. döngü: i=2
^C1. döngü: i=3
1. döngü: i=4
2. döngü:i=0
2. döngü:i=1
^C
9_Sig  ls\> 
```

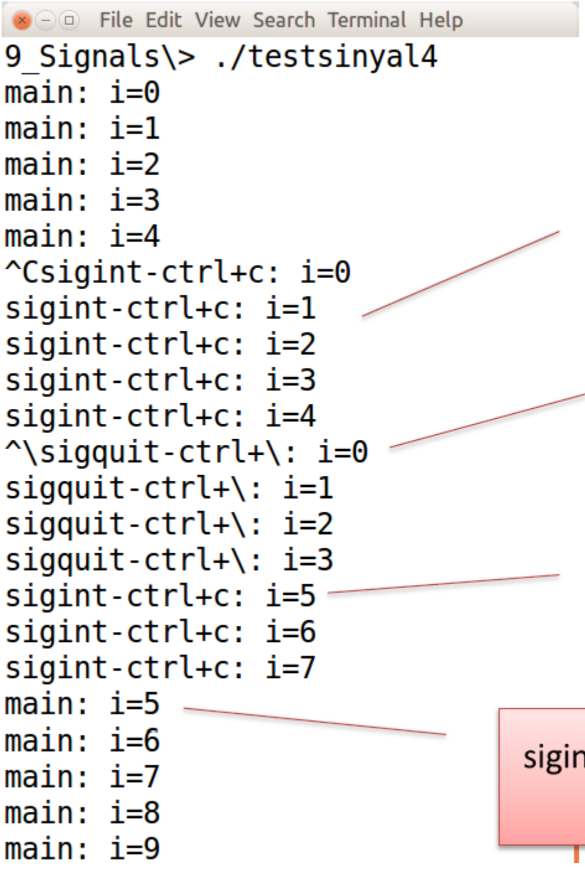
Birinci döngü çalışırken i=3 iterasyonunda ctrl+c ile gönderilen sinyal ihmal ediliyor.

İkinci döngüde ise i=1 iterasyonunda ctrl+c ile default tanımlanmış programı sonlandırma işlemi gerçekleşiyor.

Calling a signal within a signal



```c
#include <stdio.h>
#include <signal.h>
void sigint_handler(int signum)
{       int i;
        for(i=0;i<8;i++)
        {
                printf("sigint-ctrl+c: i=%d \n",i);
                sleep(1);
        }
}
void sigquit_handler(int signum)
{       int i;
        for(i=0;i<4;i++)
        {
                printf("sigquit-ctrl+\\: i=%d \n",i);
                sleep(1);
        }
}
main()
{
int i;
signal(SIGINT,sigint_handler);
signal(SIGQUIT,sigquit_handler);
for(i=0;i<10;i++)
{
        printf("main: i=%d \n",i);
        sleep(1);
}
}
```



```
9_Signals\> ./testsinyal4
main: i=0
main: i=1
main: i=2
main: i=3
main: i=4
^Csigint-ctrl+c: i=0
sigint-ctrl+c: i=1
sigint-ctrl+c: i=2
sigint-ctrl+c: i=3
sigint-ctrl+c: i=4
^\sigquit-ctrl+\: i=0
sigquit-ctrl+\: i=1
sigquit-ctrl+\: i=2
sigquit-ctrl+\: i=3
sigint-ctrl+c: i=5
sigint-ctrl+c: i=6
sigint-ctrl+c: i=7
main: i=5
main: i=6
main: i=7
main: i=8
main: i=9
```

Ctrl+c ile SIGINT oluşturuldu. main() çalışmasını durdurdu. sigint_handler() fonksiyonu çalışıyor.

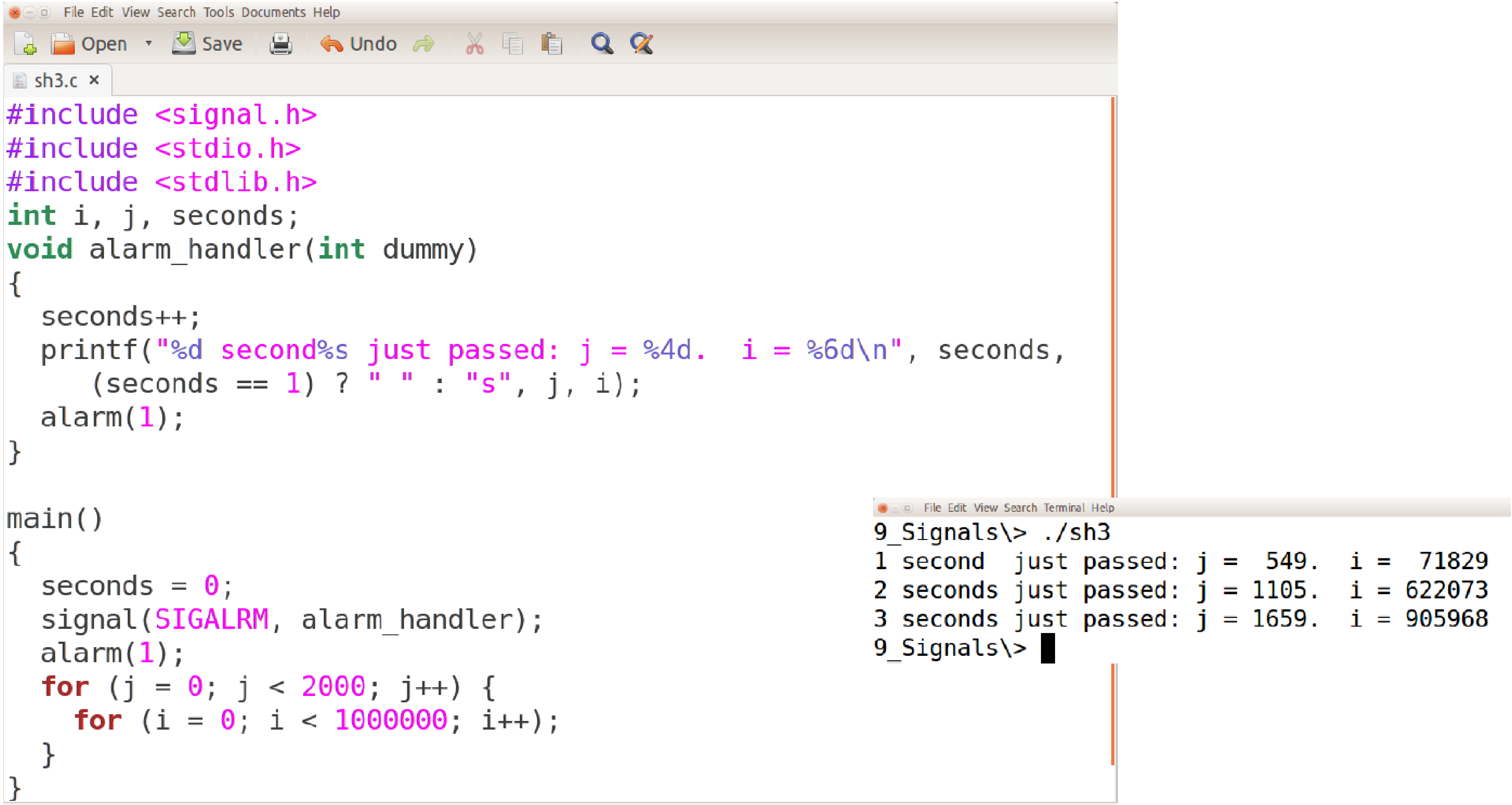
Ctrl+\ ile SIGQUIT oluşturuldu. Şimdi sigquit_handler() çalışıyor.

sigquit_handler() sonlandı. sigint_handler() kaldığı yerden devam ediyor.

sigint_handler() sonlandı. main() kaldığı yerden devam ediyor.

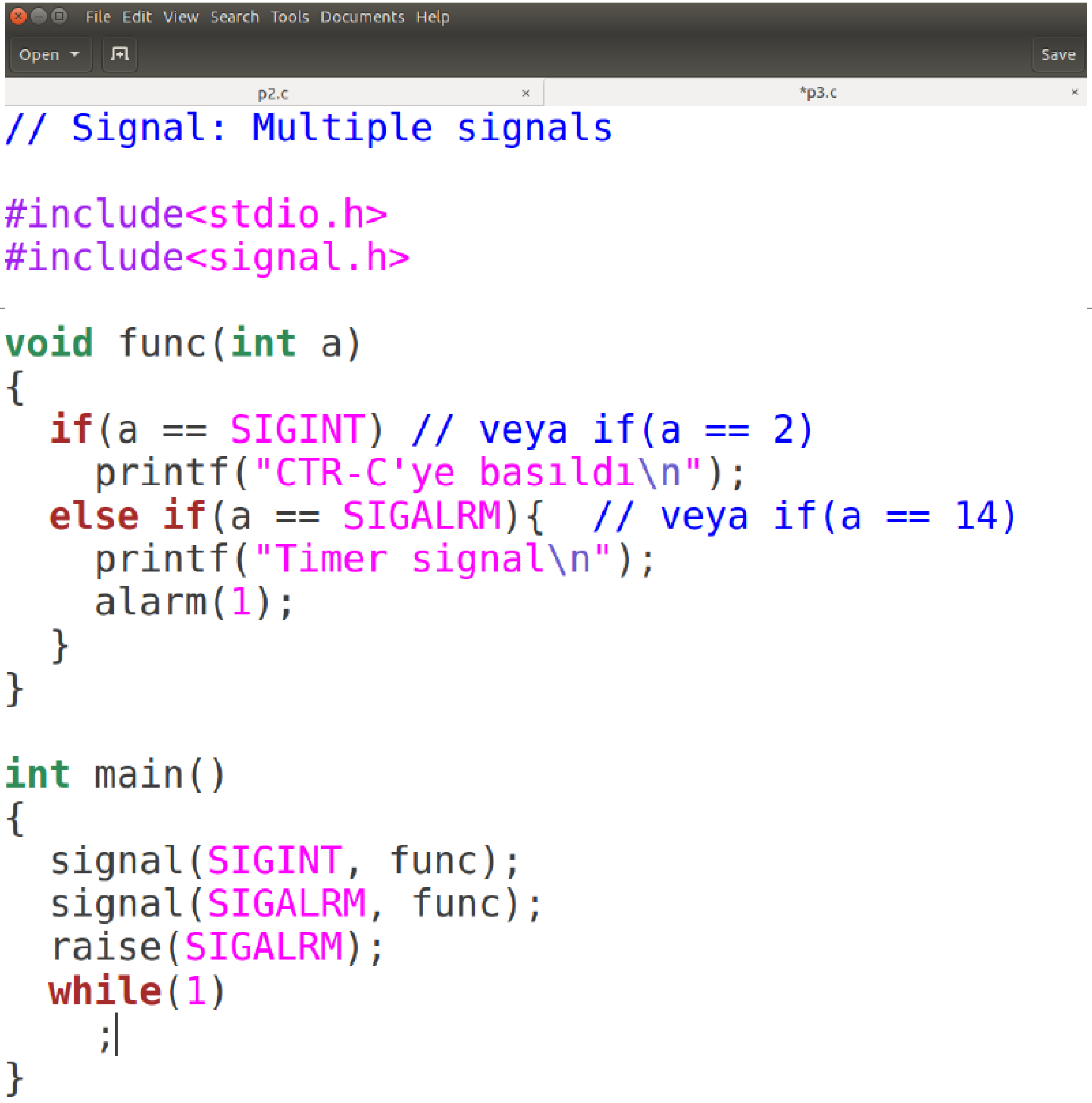
alarm-sigalarm()

- ❑With the alarm(int seconds) function, the SIGALRM signal is generated after the specified time in seconds.
- ❑In the example below, when the alarm() function in main() produces the SIGALRM signal, alarm_handler() defined in signal is called. Here, the values of the i and j variables in the main program at the moment the signal is received are printed.
- ❑Additionally, since alarm() is redefined, an alarm signal is generated at 1-second intervals as long as the loop in main() continues. If alarm(1) in alarm_handler() is removed, the alarm signal is generated only once.


File Edit View Search Tools Documents Help
Open
Save
Undo
sh3.c
File Edit View Search Terminal Help

```c
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
int i, j, seconds;
void alarm_handler(int dummy)
{
  seconds++;
  printf("%d second%s just passed: j = %4d.  i = %6d\n", seconds,
     (seconds == 1) ? " " : "s", j, i);
  alarm(1);
}

main()
{
  seconds = 0;
  signal(SIGALRM, alarm_handler);
  alarm(1);
  for (j = 0; j < 2000; j++) {
    for (i = 0; i < 1000000; i++);
  }
}
```

```
9_Signals\> ./sh3
1 second  just passed: j =  549.  i =  71829
2 seconds just passed: j = 1105.  i = 622073
3 seconds just passed: j = 1659.  i = 905968
9_Signals\> 
```



```c
// Signal: Multiple signals

#include<stdio.h>
#include<signal.h>

void func(int a)
{
  if(a == SIGINT) // veya if(a == 2)
    printf("CTR-C'ye basıldı\n");
  else if(a == SIGALRM){  // veya if(a == 14)
    printf("Timer signal\n");
    alarm(1);
  }
}

int main()
{
  signal(SIGINT, func);
  signal(SIGALRM, func);
  raise(SIGALRM);
  while(1)
    ;
}
```

Assignment

Write a program in C language that changes the default behavior when the ctrl+c key is pressed and prints a randomly generated number on the screen.