Project #0-1: Installing Pintos

[CSE4070]

Instructors

Prof. Sungyong Park

Teaching Assistants

Hongsu Byun

Fall 2021

Linux Instructions and Vim Usage

Useful Linux instructions

- man
- mkdir/rmdir
- cp
- mv
- rm
- cat
- echo
- grep
- ps
- kill
- pwd
- su/passwd
- tar

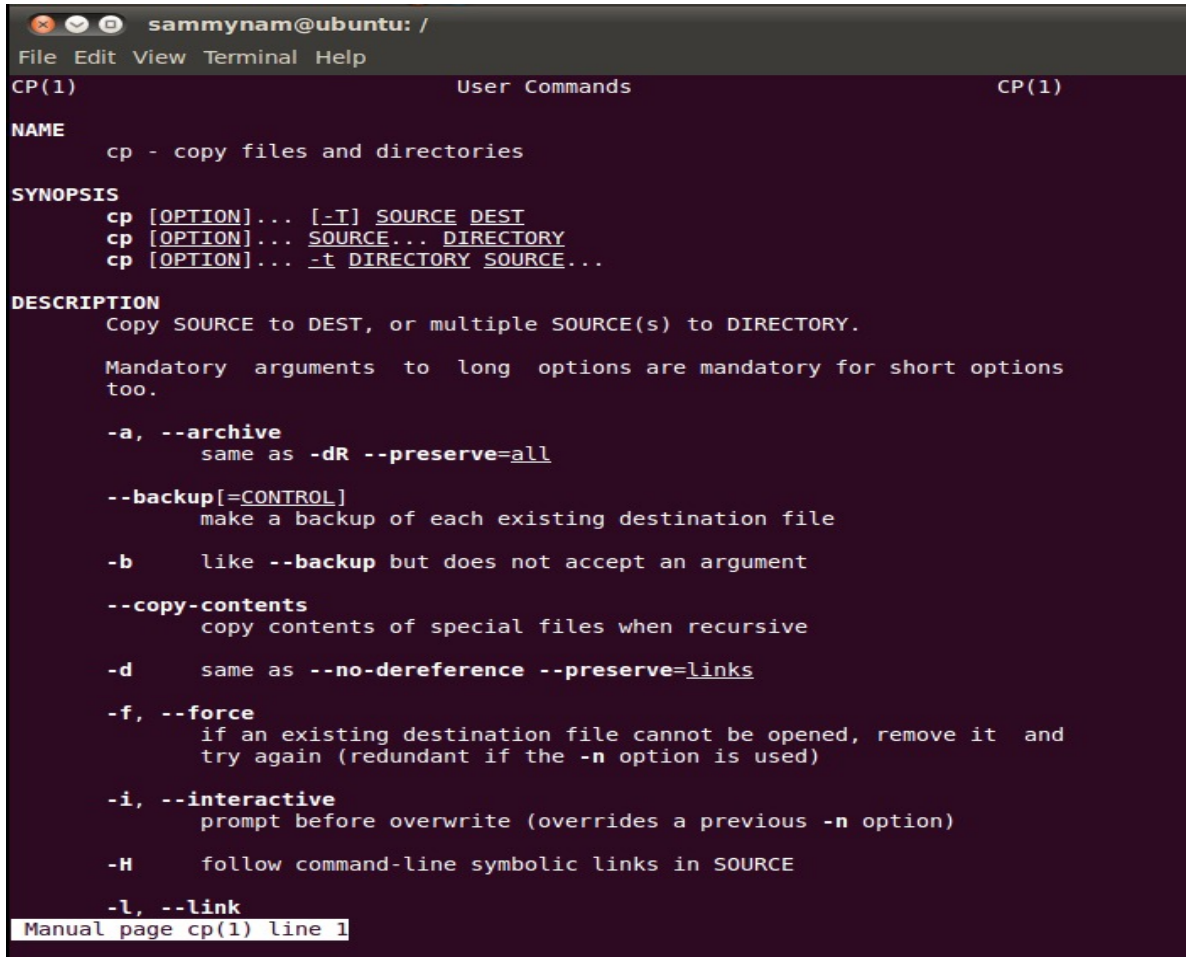
man

Provides description and usage for Linux commands

Usage: man [instruction]

Ex:

$ man cp



mkdir/rmdir

Make/remove directory

If the directory is not empty, use 'rm -r' to remove it

Usage: mkdir [option] [Directory Name]

rmdir [option] [Directory Name]

Ex:

$ mkdir temp

cp

Copy the file (Original file is preserved)

Usage: cp [option] [src] [dst]

Ex:

$ cp a.c temp

mv

Move file or rename file (Original file is disappeared)

Usage: mv [option] [src] [dst]

Ex:

$ mv a.c b.c

rm

Remove file or directory.

Usage: rm [option] [filename]

Ex:

$ rm –rf temp

If you perform 'rm -rf *' in /(root) directory, every file will be deleted from the system.

-rf option indicates recursive and force, respectively.

cat

1. Print the contents of the file on the standard output.

Usage: cat [option] [filename]

Ex:

$ cat tempfile

$ cat > test.txt (Get data from standard input; user can input data until user does [Ctrl+D])

$ cat < test.txt (Print the contents of the file)

2. Concatenate files

Ex:

$ cat test.txt test2.txt > test12.txt (Concatenate test.txt and test2.txt and make a file "test12.txt")

echo

Prints string or system environment variables.

Usage: echo [string…]

Ex:

$ echo $PATH

$ echo x

grep

Print lines matching a pattern from files or standard input.

Usage: grep [option] PATTERN [File…]

-n: print the line and line number in FILE which is matched.

-i: ignore case distinctions.

-l: print only FILE name, which contains PATTERN matched.

Ex:

$ grep –n ftp /etc/groupt

$ grep –i the /etc/init.d/qmail

$ grep –il ftp /etc/init.d/*

ps

Report the list of current processes.

Usage: ps [option]

-ef: print the all processes with full-format listing.

-au: print the username and start time of processes including other users' processes.

Ex:

$ ps -ef

$ ps -au

kill

Sends signal to the processes.

Representative signal is SIGKILL which is used to forcefully terminate a process.

Usage: kill [option] [process id]

-l: print list of signals

Ex:

$ kill -9 4914 (force quit process #4914)

```
sammynam@ubuntu:~/Desktop$ 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
```

pwd

Checks the current directory.

Usage: pwd

```
sammynam@ubuntu:~/Desktop$ pwd
/home/sammynam/Desktop
sammynam@ubuntu:~/Desktop$
```

su/passwd

su: switch user ID or become superuser.

passwd: change user password.

Usage: su [options] [username]

Ex:

$ su (if the USERNAME is omitted than it will switch the account to the superuser.)

$ passwd (change the password of current account.)

tar

Compresses or extracts file.

Usage: tar [options] [**pathname**]

-c/x: compress / Extract

-v: verbosely list the files processed

-f: use the file to compress or extract

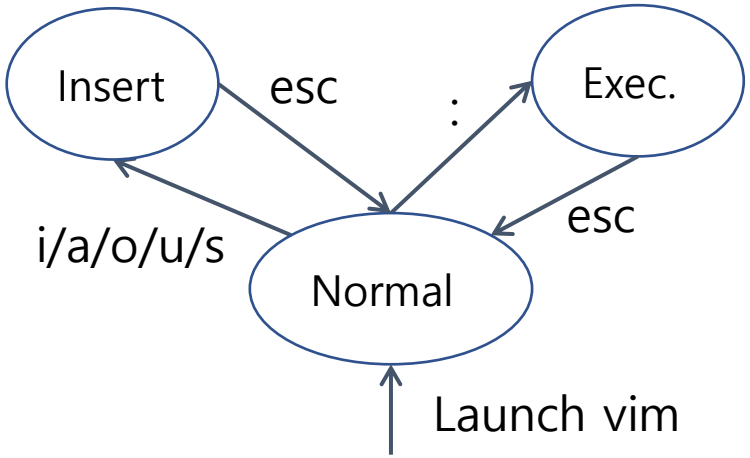
Ex)

$ tar -cvzf sample.tar.gz pintos (if pintos is a directory, it will be compressed into sample.tar.gz)

$ tar -xvzf sample.tar.gz (extract sample.tar.gz)

vim

- Vim is known as Visual Interface iMproved, which is an improved version of vi



- Normal mode:
  - yy+p: copy and paste line, /: search, x: delete character, dd: delete line, u: undo.
  - v : change into visual mode.
- Insert mode
  - i : Insert, a : Append
- Execution mode (Type ': (colon)' in normal mode)
  - w: save, q: quit, wq!: quit without saving

Pintos Installation

Caution

- **We will use CSPRO9 (cspro9.sogang.ac.kr) and CSPRO10 (cspro10.sogang.ac.kr)**
- So do not try to run Pintos on CSPRO (cspro.sogang.ac.kr) server
- Note that the CSPRO server indicates CSPRO9 or CSPRO10 from now on

Pintos & Emulator

- Pintos is simple OS framework for 80x86 architecture.
- Use system simulator that simulates an 80x86 CPU and its peripheral devices.
- Project Category : Kernel Threads, User Programs, Virtual Memory, File Systems.
  (All projects except File Systems will be covered in this class.)
- Features
  1) Support user and kernel thread.
  2) Allow running user program (basic UNIX commands like echo, ls, cat, pwd, …).
  3) Support simple file system.
  4) Implemented in C language.
  5) Well-Documented Project & Grading System.
- We will use QEMU as an emulator for Pintos.

Pintos Installation

1. Download Pintos file
   - We provide modified code in e-class, so don't use original source code from Stanford University.
2. Extract the file
   - $ tar -xvzf pintos_modified.tar.gz

✓ You don't need to install QEMU in the CSPRO sever. It is already installed.

Pintos Installation

- Before running Pintos, we need to setup .bashrc file in the home directory
  1. Open ~/.bashrc with editor.
  2. Add the following line at the end of the file:
     export PATH=/sogang/under/<YOUR_ACCOUNT>/pintos/src/utils:$PATH (학부생)
     export PATH=/sogang/grad/<YOUR_ACCOUNT>/pintos/src/utils:$PATH (대학원생)
  3. Run the following command to apply the changes in bash shell:
     $ source ~/.bashrc

```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH=/sogang/under/cse20189999/pintos/src/utils:$PATH
```

Running Pintos

- Build Pintos (assume that you have already extracted the file on your home directory)
  - $ cd ~/pintos/src/threads
  - $ make
  - Consequently, 'build' directory will be created in the current directory (src/threads).
- Run Pintos
  - Pintos provides 'pintos' utility that helps running Pintos by QEMU.
  - 'pintos' utility is in src/utils.
  - **Go to src/threads** and run the following command (you should run it in src/threads, not src/utils).

    ~/pintos/src/threads $ ../utils/pintos -v -- -q run alarm-multiple
    Or
    ~/pintos/src/threads $ pintos -v -- -q run alarm-multiple
    (Note that you should input one space among '-v', '--' and '-q')

```
(alarm-multiple) thread 4: duration=50, iteration=6, product=300
(alarm-multiple) thread 4: duration=50, iteration=7, product=350
(alarm-multiple) end
Execution of 'alarm-multiple' complete.
Timer: 580 ticks
Thread: 0 idle ticks, 581 kernel ticks, 0 user ticks
Console: 2954 characters output
Keyboard: 0 keys pressed
Powering off...
cse20189999@cspro10:~/pintos/src/threads$
```

Running Pintos

- If you face the error like below, check the current directory where you run pintos.
- Since the current directory is src/utils, Pintos cannot find its kernel and error occurs.
- If you execute Pintos in src/threads, Pintos will find the kernel in src/threads/build/kernel.bin

```
cse20189999@cspro10:~/pintos/src/utils$ pintos -v -- -q run alarm-multiple
Cannot find kernel
cse20189999@cspro10:~/pintos/src/utils$ cd ../threads
cse20189999@cspro10:~/pintos/src/threads$ pintos -v -- -q run alarm-multiple
```

Project Test

- Each project has its own test program
  - Test program is in src/tests.
  - You can use this program to test your implementation by yourself.
  - For example, in project 3, you can test by running 'make check' in src/threads/
    - ~/pintos/src/threads $ make check
    - PASS/FAIL will be printed for each test case

```
pass tests/threads/alarm-single
pass tests/threads/alarm-multiple
pass tests/threads/alarm-simultaneous
FAIL tests/threads/alarm-priority
pass tests/threads/alarm-zero
pass tests/threads/alarm-negative
FAIL tests/threads/priority-change
FAIL tests/threads/priority-donate-one
FAIL tests/threads/priority-donate-multiple
FAIL tests/threads/priority-donate-multiple2
```

※ src/threads/build/results

Requirements

Project#0-1

1. In **CSPRO9 or CSPRO10 (not CSPRO)** server, run **$pintos -v -- -q run alarm-multiple** and **capture the result of it** (you can just capture the last few lines of the result, **but your ID should be shown in the capture**).
2. **If your ID is not shown in the capture, deduct 10 points.**
   If "Powering off..." is not shown in the capture, deduct 70 points.
   If the Pintos doesn't quit properly (Kernel panic or other errors), deduct 70 points.
3. Use your own account in the server. (Don't borrow other's account.)
4. **Due Date: 9/19 23:59**
   Late submission is allowed up to 3 days (~9/22) and **10% of point will be deducted per day**
5. **Submit the capture file on e-class website**
   (Please use .jpg or .png extensions. Do not use other formats.)
6. **File name should be the following form:**
   os#0_1_ID#.jpg or os#0_1_ID#.png
   e.g.) os#0_1_20171234.jpg or os#0_1_20175678.png

Project#0-1

7. No hardcopy.

Reference Homepages

pintos	http://www.stanford.edu/class/cs140/projects/index.html
pintos document	http://www.stanford.edu/class/cs140/projects/pintos/pintos.pdf