

Operating Systems :

Linux instructions & vim usage



CSPRO 계정

- ID : cse + <자신의 학번>
- P/W : os2016

- 계정 등록이 안되어 있는 수강생은 해당 분반 조교에게 메일바랍니다.

Useful Linux instructions

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

man

리눅스에서 사용되는 명령어들에 대한 설명과 용법들을 제공.

Usage) man [instruction]

Ex)

#man grep

```
sammynam@ubuntu: /
File Edit View Terminal Help
CP(1) User Commands CP(1)

NAME
  cp - copy files and directories

SYNOPSIS
  cp [OPTION]... [-I] SOURCE DEST
  cp [OPTION]... SOURCE... DIRECTORY
  cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
  Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --archive
      same as -dR --preserve=all

  --backup[=CONTROL]
      make a backup of each existing destination file

  -b      like --backup but does not accept an argument

  --copy-contents
      copy contents of special files when recursive

  -d      same as --no-dereference --preserve=links

  -f, --force
      if an existing destination file cannot be opened, remove it and
      try again (redundant if the -n option is used)

  -i, --interactive
      prompt before overwrite (overrides a previous -n option)

  -H      follow command-line symbolic links in SOURCE

  -l, --link

Manual page cp(1) line 1
```

mkdir/rmdir

디렉토리를 생성/제거한다.

디렉토리를 제거하기 위해서는 디렉토리안에 어떤 파일도 있어선 안됨

Usage) mkdir [option] [Directory Name]

Ex)

#mkdir temp (상대경로)

Or

#mkdir ./temp (절대경로)

cp

파일을 복사

복사 시 원본 파일은 그대로 남아있음

Usage) cp [option] [src] [dst]

Ex)

#cp a.c temp (상대경로)

Or

#cp ./a.c ./temp (절대경로)

mv

파일이동을 위해 사용
이동시 원본파일이 삭제됨
파일의 이름을 바꿀 때도 사용

Usage) mv [option] [src] [dst]

Ex)

#mv a.c b.c (상대경로)

Or

#mv ./a.c ./b.c (절대경로)

rm

파일이나 디렉토리를 삭제

Usage) `rm [option] [filename]`

Ex)

`#rm -rf temp` (상대경로)

Or

`#rm -rf ./temp` (절대경로)

최상위폴더(/)에서 **rm -rf *** 을 하면 모든 데이터가 삭제될 수 있음

항상 데이터를 지울 때는 조심할 것

rf 옵션은 **reculsive**와 **force**의 약자로 데이터를 지울지 묻지않고 지움
파일이 존재하는 디렉토리를 지운다고 할지라도 그 안의 내용도 지움

cat

1. 파일의 내용을 표준출력으로 보내줌

Usage) cat [option] [filename]

Ex)

#cat tempfile

#cat > test.txt (표준입력으로 데이터 받기; 사용자가 Ctrl + D를 입력할 때까지 데이터입력가능)

#cat < test.txt (파일내용확인)

2. 파일병합

파일 test.txt와 test2.txt파일을 병합하여, test12.txt.파일로 생성

#cat test.txt test2.txt > test12.txt

echo

인자(argument) 문장 혹은 시스템 환경 변수 등을 출력해 주는 명령어

Usage) echo [문자열...]

Ex)

#echo \$PATH (현재 PATH를 출력)

#echo x

grep

파일이나 표준입력으로부터 패턴 혹은 문장을 입력하여, 검색하려는 파일 혹은 출력에서 동일한 값을 검색

Usage) `grep [option] PATTERN [File...]`

- n : 파일안에서 발견된 매치되는 라인과 그 라인 수를 출력함
- i : 대소문자 구분없이 패턴 출력
- li : 패턴이 들어있는 파일만 출력. (여러파일에서 검색할때 용이)

Ex)

```
#grep -n ftp /etc/groupt
#grep -I the /etc/init.d/qmail
#grep -il ftp /etc/init.d/*
```

ps

현재 구동되고 있는 프로세스 리스트를 보여줌

Usage) ps [option]

ps aux : 다른 사용자가 실행하고 있는 프로그램의 목록까지 보여줌.

ps -ef

ps -au

Pstree

현재 실행되고있는 프로세스를 트리형태로 보여줌.

kill

Signal을 프로세스에 보내는 명령어.

Signal의 종류는 여러가지가 있으며 대표적인 signal로는 SIGKILL(강제종료)가 있으며 이는 프로세스를 강제로 종료할 때 사용됨

Usage) kill [option] [process id]

-l : signal의 목록을 출력

Ex) #kill -9 4914 (4914번 프로세스를 강제 종료함)

```
sammyam@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

현재 자신이 있는 디렉터리를 확인하고자 할 경우 사용됨
보통 경로명을 **PATH**에 지정할 때 경로를 보기위해 사용함

Usage) pwd

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

su/passwd

su : 슈퍼유저(root) 권한을 얻기위해 사용

passwd : 계정의 패스워드 변경을 할 때 사용

Ex)

#su

#sudo passwd root (root계정의 비밀번호를 변경한다.)

```
sammynam@ubuntu:~/Desktop$ sudo passwd root
[sudo] password for sammynam:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
sammynam@ubuntu:~/Desktop$
```

tar

리눅스에서 파일을 압축하거나 해제할 때 쓰는 명령어

#tar cvfz 파일명.tar.gz [파일1] [파일2] ... : 묶고 압축하기

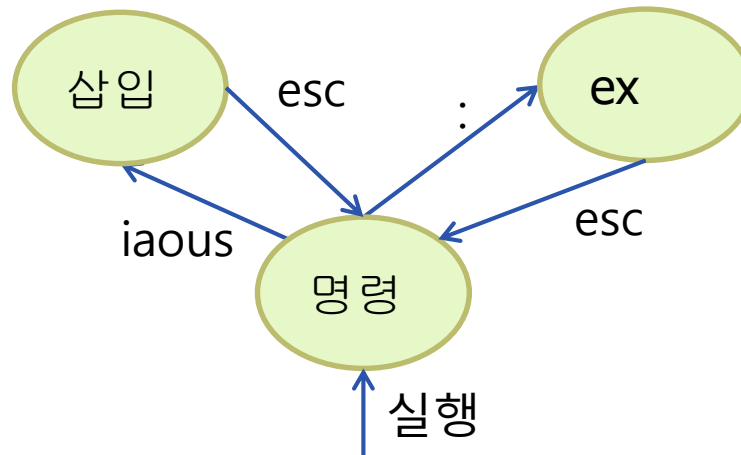
#tar xvfz 파일명.tar.gz : 묶고 압축된 파일 풀기

tar는 여러 개 파일을 하나로 묶는 명령어.

gz는 실질적으로 압축

vim

Vim(Visual Interface iMproved)의 약자로 vi의 성능을 개선시킨 vi의 차기 버전



Vim에서 삽입모드에서 자주사용되는 몇가지 유용한 명령어

명령모드

yy+p : 라인복사 , **/** : 검색 , **x** : 글자삭제 , **dd** : 라인삭제 , **u** :이전상태로 되돌리기 ,
v : **visual**모드로 변경. (변경된상태에서 커서를 움직여 블록지정후 **y**로 복사 **p**로 붙이기를 할 수 있다.)

삽입모드

I : 삽입, **a** : 덧붙임

Ex모드

:w :저장 , **:q** 종료, **:wq!** :저장하지 않고 종료

Operating Systems :

Project #0-1. Pintos 소개 및 설치

Pintos & Emulator

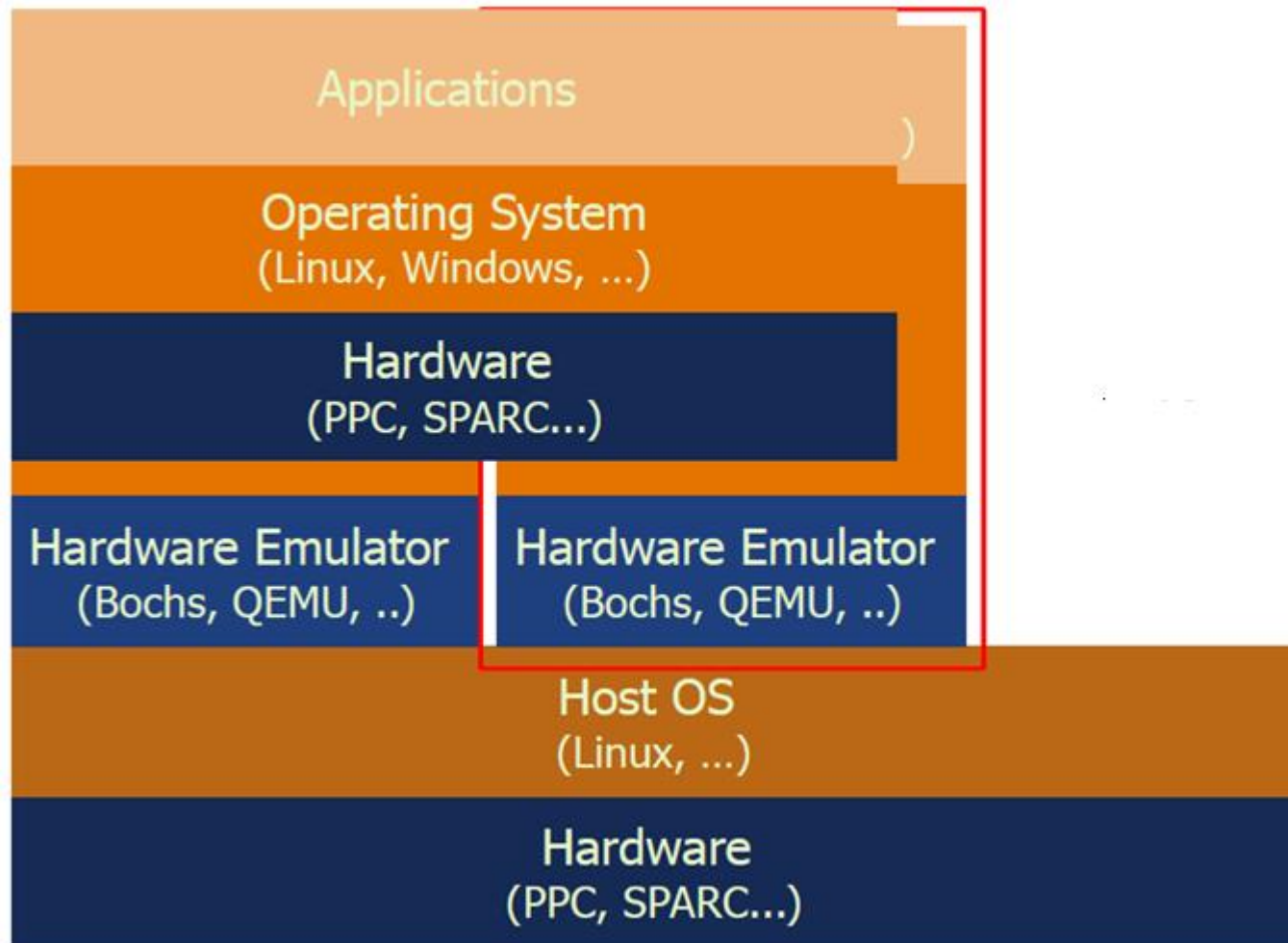
Pintos

- Simple OS Framework for 80x86 architecture
- Use system simulator that simulates an 80x86 CPU and its peripheral devices
- Project Category : kernel thread, user program, virtual memory, file system
- Features
 1. support user and kernel thread
 2. allow running user program(basic unix commands like echo, ls, cat, pwd, ...)
 3. support file system
 4. implemented in C language
 5. Well-Documented Project & Grading System

Bochs & qemu

- Open source IA-32(x86) PC emulator
- Implemented in C++
- Use this program for OS project in our class

Emulator의 구동



Linux – 설치(Ubuntu 10.04기준)

1. Pintos 다운로드 및 압축해제

- 과목 homepage에서 자체 제공(스탠포드 original source 사용 금지.)
→ `pintos_modified.tar`
- `$tar xvfz pintos_moidified.tar.gz`
- 이후 환경변수 설정 등은 `pintos manual` 참조(보조자료)
(Appendix G. Installing Pintos)

2. Qemu Install(Ubuntu 10.04 기준 & recommended)

- `$sudo apt-get install qemu`
- `sudo` : 명령에 관리자 권한 부여(관리자 계정의 경우 필요 없음)
- `apt-get` : 저장소 접근 명령
- `install` : 설치 명령
- `qemu` : 프로그램 명.
- `bochs`와 달리 `qemu`의 경우 10.04에서 제공하는 최신 버전을 사용하면 됨

Linux – 설치(Ubuntu 10.04기준)

Default emulator를 qemu로 변경

- threads/, userprog/, filesys/, vm/ 각각 안의 Make.vars
 - SIMULATOR = --bochs
 - + SIMULATOR = --qemu
- utils/pintos(혹은 실행파일 설치 경로상의 pintos) : 실행파일
line 103 :
 - \$sim = "bochs" if !defined \$sim;
 - + \$sim = "qemu" if !defined \$sim;
line 624 :
 - push (@cmd, '-no-kqemu');
 - + #push (@cmd, '-no-kqemu');
line 129, 130(maybe not necessary)
 - --bochs (default) Use Bochs as simulator
 - --qemu Use QEMU as simulator
 - + --qemu (default) Use QEMU as simulator
 - + --bochs User Bochs as simualtor

Linux – Running Pintos

1. Pintos 압축 푼 디렉토리의 src로 이동

- `$cd thread`
- `$make` : 결과로 현재 디렉토리에 (src/thread) build라는 디렉토리 생성됨

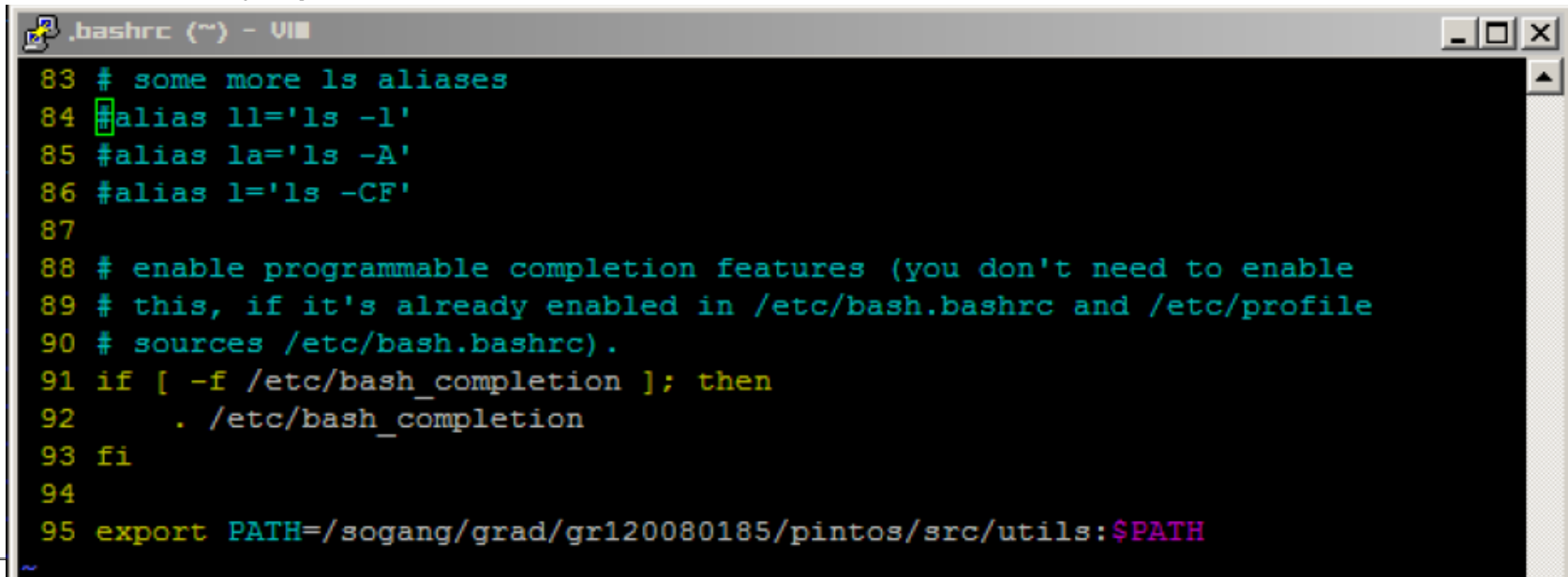
2. pintos 실행

- `pintos` 는 Pintos를 시뮬레이터에서 수행하는데 편의를 제공하기 위해 만든 파일로 `src/utls`에 있음
- `$../utls/pintos run alarm-multiple`

CSPRO – 설치

I. CSPRO I~2에 접속

- BOCHS, QEMU 이미 설치되어 있음
- Pintos/src/utils 안의 스크립트 및 실행파일 사용
- 에디터를 이용하여 **.bashrc** 파일 오픈. (**vi ~/.bashrc**)
 - ‘export PATH=/sogang/under/<자신의 계정(cs학번)/pintos/src/utils:\$PATH’ 추가



```
.bashrc (~) - vi
83 # some more ls aliases
84 #alias ll='ls -l'
85 #alias la='ls -A'
86 #alias l='ls -CF'
87
88 # enable programmable completion features (you don't need to enable
89 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
90 # sources /etc/bash.bashrc).
91 if [ -f /etc/bash_completion ]; then
92     . /etc/bash_completion
93 fi
94
95 export PATH=/sogang/grad/gr120080185/pintos/src/utils:$PATH
```



CSPRO – 설치

Default emulator를 qemu로 변경

- threads/, userprog/, filesys/, vm/ 각각 안의 Make.vars
 - SIMULATOR = --bochs
 - + SIMULATOR = --qemu
- utils/pintos(혹은 실행파일 설치 경로상의 pintos) : 실행파일
line 103 :
 - \$sim = "bochs" if !defined \$sim;
 - + \$sim = "qemu" if !defined \$sim;
line 624 :
 - push (@cmd, '-no-kqemu');
 - + #push (@cmd, '-no-kqemu');
line 129, 130(maybe not necessary)
 - --bochs (default) Use Bochs as simulator
 - --qemu Use QEMU as simulator
 - + --qemu (default) Use QEMU as simulator
 - + --bochs User Bochs as simualtor

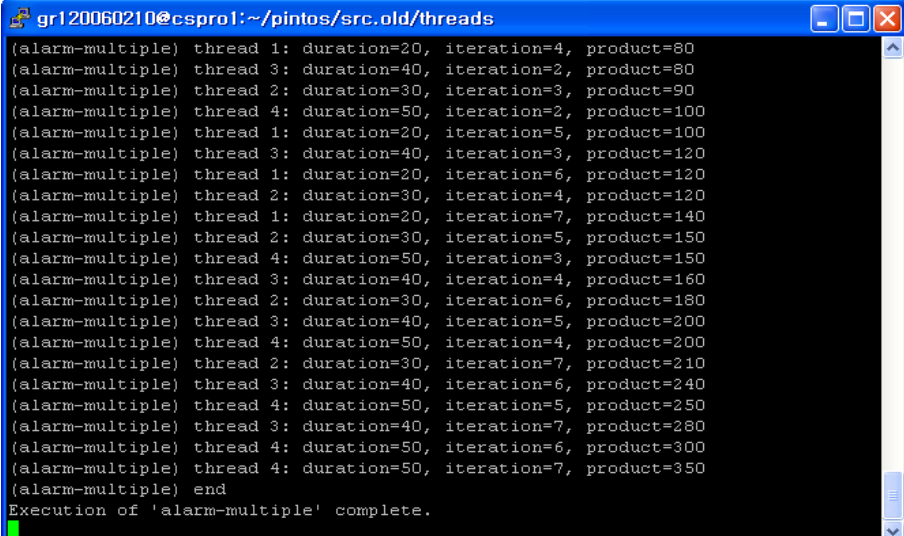
CSPRO – 설치

1. Pintos 압축 툰 디렉토리의 src로 이동

- `$cd thread`
- `$make` : 결과로 현재 디렉토리에 (src/thread) build라는 디렉토리 생성됨

2. pintos 실행

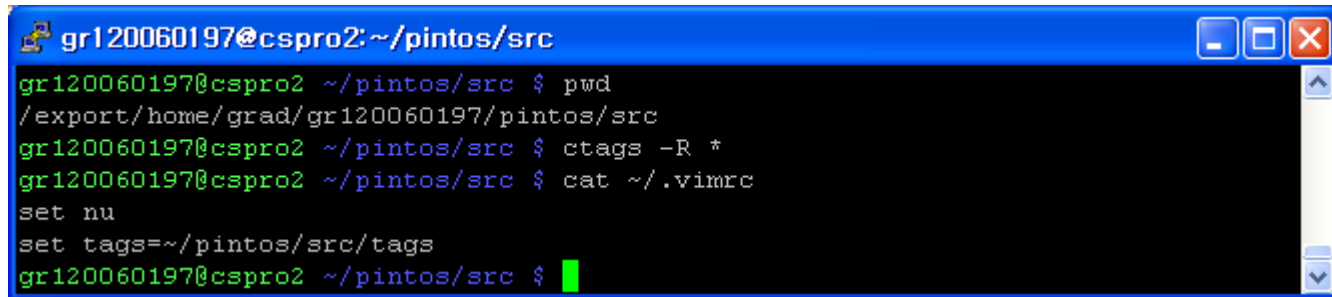
- pintos 는 Pintos를 시뮬레이터에서 수행하는데 편의를 제공하기 위해 만든 파일로 src/utils에 있음
- `$../utils/pintos -v -- -q run alarm-multiple` 또는
`pintos -v -- -q run alarm-multiple`



```
gr120060210@csp1:~/pintos/src.old/threads
(alarm-multiple) thread 1: duration=20, iteration=4, product=80
(alarm-multiple) thread 3: duration=40, iteration=2, product=80
(alarm-multiple) thread 2: duration=30, iteration=3, product=90
(alarm-multiple) thread 4: duration=50, iteration=2, product=100
(alarm-multiple) thread 1: duration=20, iteration=5, product=100
(alarm-multiple) thread 3: duration=40, iteration=3, product=120
(alarm-multiple) thread 1: duration=20, iteration=6, product=120
(alarm-multiple) thread 2: duration=30, iteration=4, product=120
(alarm-multiple) thread 1: duration=20, iteration=7, product=140
(alarm-multiple) thread 2: duration=30, iteration=5, product=150
(alarm-multiple) thread 4: duration=50, iteration=3, product=150
(alarm-multiple) thread 3: duration=40, iteration=4, product=160
(alarm-multiple) thread 2: duration=30, iteration=6, product=180
(alarm-multiple) thread 3: duration=40, iteration=5, product=200
(alarm-multiple) thread 4: duration=50, iteration=4, product=200
(alarm-multiple) thread 2: duration=30, iteration=7, product=210
(alarm-multiple) thread 3: duration=40, iteration=6, product=240
(alarm-multiple) thread 4: duration=50, iteration=5, product=250
(alarm-multiple) thread 3: duration=40, iteration=7, product=280
(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.
```

Tags(I)

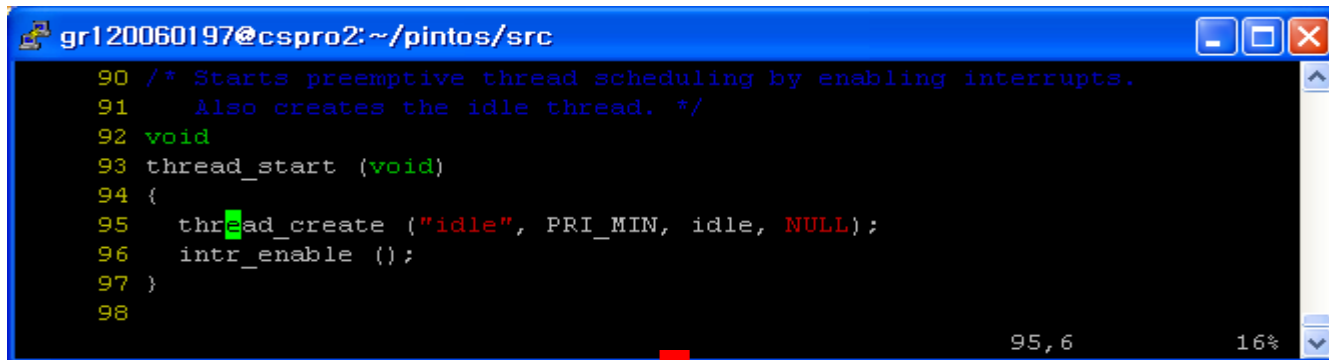
1. 함수나 전역변수에 쉽게 이동할 수 있는 인덱스 파일
2. Vi와 연동해서 사용할 수 있다.
3. Make `tags` file & set for VI



```
gr120060197@csp2:~/pintos/src
gr120060197@csp2 ~/pintos/src $ pwd
/export/home/grad/gr120060197/pintos/src
gr120060197@csp2 ~/pintos/src $ ctags -R *
gr120060197@csp2 ~/pintos/src $ cat ~/.vimrc
set nu
set tags=~/pintos/src/tags
gr120060197@csp2 ~/pintos/src $
```

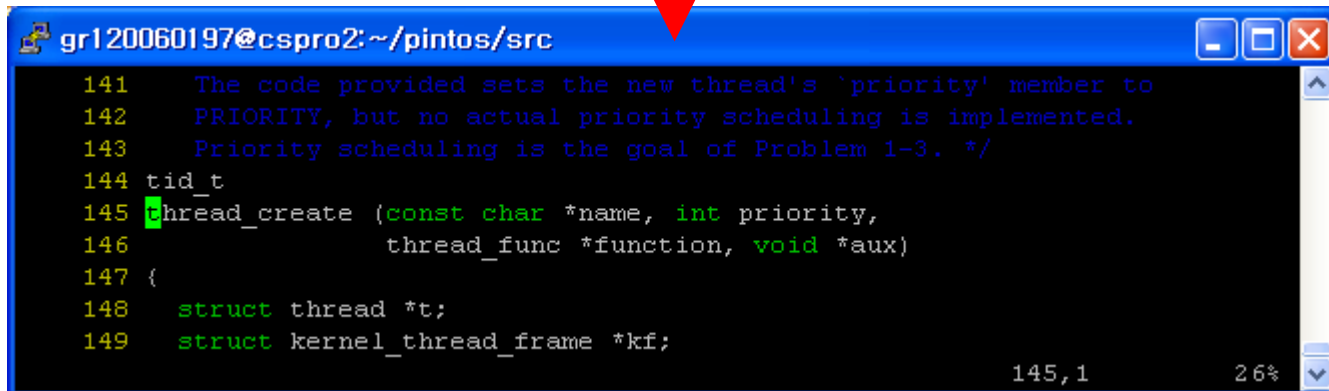
Tags(2)

1. 이동하고자 하는 함수나 전역변수에 커서를 두고 Ctrl+]
2. 바로 전 단계로 이동하고자 할 때는 Ctrl+T



A terminal window titled 'gr120060197@cspro2:~/pintos/src'. The code displayed is from the file 'thread.c'. Line 95, 'thread_create ("idle", PRI_MIN, idle, NULL);', has a green cursor at the start of the string 'idle'. The status bar at the bottom right shows '95,6' and '16%'.

```
gr120060197@cspro2:~/pintos/src
90 /* Starts preemptive thread scheduling by enabling interrupts.
91    Also creates the idle thread. */
92 void
93 thread_start (void)
94 {
95   thread_create ("idle", PRI_MIN, idle, NULL);
96   intr_enable ();
97 }
98
```



A terminal window titled 'gr120060197@cspro2:~/pintos/src'. The code displayed is from the file 'thread.c'. Line 145, 'thread_create (const char *name, int priority, thread_func *function, void *aux)', has a green cursor at the start of the string 'const'. The status bar at the bottom right shows '145,1' and '26%'.

```
gr120060197@cspro2:~/pintos/src
141 The code provided sets the new thread's 'priority' member to
142 PRIORITY, but no actual priority scheduling is implemented.
143 Priority scheduling is the goal of Problem 1-3. */
144 tid_t
145 thread_create (const char *name, int priority,
146               thread_func *function, void *aux)
147 {
148   struct thread *t;
149   struct kernel_thread_frame *kf;

```

GDB(I)

1. Pintos는 기존의 GDB에 기능을 추가하여 편리한 스크립트를 함께 제공한다.
2. --gdb 옵션과 함께 pintos 를 실행하며, 또 다른 터미널의 build 디렉토리에서 pintos-gdb를 실행한다.
 - EX> \$pintos -gdb -run alarm-multiple
3. Gdb의 기본적인 사용법은 널리 알려진 문서를, Pintos를 위한 스크립트는 Appendix E.5를 참고.

GDB(2)

1. Pintos-gdb는 gdb의 remote debug를 이용하므로, 각 사용자가 유일한 port 번호를 통해 접근해야 한다.
2. 따라서 각자 자신만의 port 번호로 설정한다.
3. Utils/pintos line 566
print BOCHSRC “gdbstub: enabled=1\n” if \$debug eq ‘gdb’;
-> print BOCHSRC “gdbstub: enabled=1, port=xxxxx\n” if
\$debug eq ‘gdb’;
(xxxxx는 바꾸고나 하는 port 번호. 중복되지 않는 적당한
5자리 수로 설정. 65535 이하)

GDB(3)

1. Misc/gdb-macros line 163

target remote localhost:1234

-> target remote localhost:xxxxx

(pintos 파일 수정 시 선택한 port 번호와 동일하게)

2. Utils/pintos-gdb line 4

GDBMACROS=/usr/class/cs140/pintos/pintos/src/misc/gdb-macros

-> GDBMACROS=/sogang/under/(cs학번)/pintos/src/misc/gdb-macros(자신의 gdb-macros 위치를 정확히 지정)

```
gr120060197@cspro2:~/pintos/src/threads
gr120060197@cspro2 ~/pintos/src/threads $ pwd
/export/home/grad/gr120060197/pintos/src/threads
gr120060197@cspro2 ~/pintos/src/threads $ pintos --gdb -- -q run alarm-multiple
Writing command line to /tmp/s6EO1PATla.dsk...
squish-pty bochs -q
=====
                Bochs x86 Emulator 2.2.6
        Build from CVS snapshot on January 29, 2006
=====
000000000000i[      ] reading configuration from bochsrc.txt
000000000000i[      ] Enabled gdbstub
000000000000i[      ] installing nogui module as the Bochs GUI
000000000000i[      ] using log file bochsout.txt
Waiting for gdb connection on localhost:1234
Pintos booting with 4,096 kB RAM...
Kernel command line: -q run alarm-multiple
373 pages available in kernel pool
373 pages available in user pool.
Calibrating timer... 102,400 loop
Boot complete.
Executing 'alarm-multiple':
(alarm-multiple) begin
(alarm-multiple) Creating 5 thread
(alarm-multiple) Thread 0 sleeps 1
(alarm-multiple) thread 1 sleeps 2
(alarm-multiple) If successful, pr
(alarm-multiple) sleep duration wi
```

```
gr120060197@cspro2:~/pintos/src/threads/build
gr120060197@cspro2 ~/pintos/src/threads/build $ pwd
/export/home/grad/gr120060197/pintos/src/threads/build
gr120060197@cspro2 ~/pintos/src/threads/build $ pintos-gdb kernel.o
GNU gdb 6.3
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "i686-pc-linux-gnu"...Using host libthread_db library "/lib/libthread_db.so.1".

gdb> debugpintos
0x0000ffff in ?? ()
gdb> b timer_sleep
Breakpoint 1 at 0xc0103d72: file timer.c, line 93.
gdb> c

Breakpoint 1, timer_sleep (ticks=0xc0101003) at timer.c:93
93      {
gdb> d 1
gdb> c
Watchdog has expired.  Target detached.
gdb> 
```

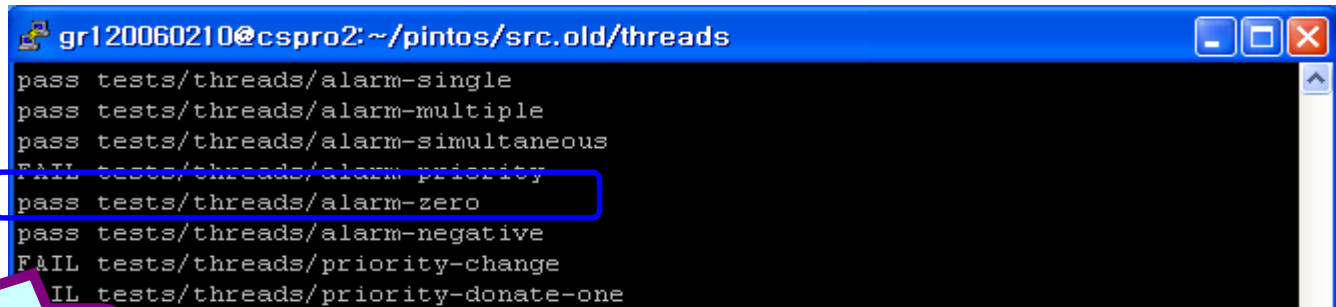

Project Testing

I. 각 프로젝트마다 test 프로그램 존재

- src/tests 에 test 프로그램 있음
- 자신이 구현한 부분에 대해 각자 테스트 가능함
- EX> src/threads/에서
 - \$make check
 - 프로젝트#1의 thread 관련 test를 수행
 - 각 test 항목마다 PASS/FAIL 출력됨
- 프로젝트의 채점 시 기준이 됨

alarm-single 항목
테스트 **PASS**

priority-change 항목
테스트 **FAIL**



```
gr120060210@cspro2:~/pintos/src.old/threads
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
IL tests/threads/priority-donate-one
```

Project#0- I

1. Pintos 설치 후

- `$pintos -v -- -q run alarm-multiple` 수행 결과를 자신의 cspro 계정에서 실행한 화면으로 캡처하여 제출
- cspro 계정 생성이 안된 수강생은 리눅스에 올린 pintos 실행 화면을 캡처하여 제출

2. 개별 과제로 제출

3. Due Date: 10월2일(일요일) 24:00 – Softcopy
10월3일(월요일) 17:00 – Hardcopy

4. Softcopy : oshw2016@gmail.com 로 제출 (양식 불일치시 0점)
- 메일 제목 : os#0_1_분반_학번 (e.g. 2반 소속 20121234일 경우, os#0_1_2_20121234)
 - 파일명 : os#0_1_분반_학번.jpg
5. Hardcopy : AS712A앞 제출함으로 제출 (분반/학번/이름 기재)

참고

Homepage

- Pintos: <http://www.stanford.edu/class/cs140/projects/index.html>
- Pintos doc:
<http://www.stanford.edu/class/cs140/projects/pintos/pintos.pdf>
- Bochs: <http://bochs.sourceforge.net>