소스코드 분석환경 구축방법
- 리눅스 커널 소스를 중심으로
(vim+plugin, ctags+cscope)

2013.12

공기석

목차

- 리눅스 커널 소스 다운로드 및 설치
- ctags + cscscope 설치
- vim 플러그인 다운로드 및 환경설정
- 소스코드 분석 환경 툴 둘러보기

리눅스 커널 소스 다운로드 및 설치

- 커널 소스 다운로드 하기
  - http://www.kernel.org/pub 로 가서 linux → kernel → v2.6 으로 이동 후 linux-2.6.30.4를 클릭
  - (command line에서의 명령)

    wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.30.4.tar.gz

- 소스 설치하기
  - mkdir ~/src
  - mv linux-2.6.30.4.tar.gz ~/src
  - cd ~/src
  - tar xvfz linux-2.6.30.4.tar.gz
  - 생성된 디렉토리와 파일의 숫자를 tree 명령으로 확인

ctags + cscope 설치하기 (1)

- ctags 를 사용한 ARM CPU용 태그 파일 생성
  - `cd ~/src/linux-2.6.30.4`
  - scripts 디렉토리 내에 tags.sh 파일이 존재하는 지 확인
    - `ls -l ./scripts/tags.sh`
  - `make tags ARCH=arm`
  - tags 파일 생성여부 확인: `ls -alh tags`
- 태그 파일의 구조
  - 심벌의 예: `start_kernel`
    - `vi +922476 tags`
  - tag_name<TAB>file_name<TAB>ex_cmd;”<TAB>extension_fields
  - tags 파일 필드의 의미

필드 이름	설명
tag_name	심볼 이름
file_name	심볼이 위치한 파일 이름
ex_cmd	파일에서 심볼을 찾을 때 사용하는 vim의 ex 모드에서 검색어 패턴의 정규식
extension_fields	심볼의 타입, f: 일반 C 함수, c: 클래스 d: define된 값

ctags + cscope 설치하기 (2)

- cscope 를 사용한 ARM용 cscope 데이터베이스 생성
  - `cd ~/src/linux-2.6.30.4`
  - `make cscope ARCH=arm`
  - `ls -alh cscope.*`
- cscope 파일들

파일 이름	설명
cscope.files	분석할 소스 파일 리스트를 포함하고 있는 파일
cscope.out	함수의 위치, 함수 콜, 매크로 변수, 전처리 심볼들에 대한 심볼 크로스 레퍼런스 파일로서 실제 데이터베이스 임
cscope.in	-q 옵션으로 데이터베이스를 생성할 때 만들어지는 파일로 심볼 검색 속도 향상을 위한 역 인덱스 (inverted index)파일
cscope.po.out	-q 옵션으로 데이터베이스를 생성할 때 만들어지는 파일로 심볼 검색 속도 향상을 위한 역 인덱스 (inverted index)파일. cscope.in과 cscope.po.out 파일 두 개가 생성됨

vim 플러그인 다운로드 하기

- 소스 분석을 위해 필요한 플러그인
  - Source Explorer
  - NERD Tree
  - Tag List
- 다운로드 사이트: www.vim.org
  - 좌측의 Scripts 링크를 클릭 → Browse all 링크 클릭
  - 하단 검색창에서 플러그인 이름을 입력하여 검색
  - 검색 결과 창에서 필요 버전을 선택하여 다운로드
    - Source Explorer: v4.3
    - NERD Tree: v4.1.0
    - Tag List: v4.5
  - 압축파일 형태의 플러그인은 압축을 풀어 ~/.vim/plugin 밑으로 복사 (압축해제 시 unzip 명령 사용)
    - `ls ~/.vim/plugin`

      `NERD_tree.vim  srcexpl.vim  taglist.vim`

vim 환경 설정 (.vimrc) (1)

- ~/.vimrc 을 ctags 와 cscope 데이터베이스 및 플러그인들과 연동할 수 있도록 수정함

```
"-------------------------------------------------------------------------------
" ctags database path 설정
"-------------------------------------------------------------------------------
    set tags=./tags,tags,/usr/src/linux-2.6.30.4/tags    "ctags DB위치

"-------------------------------------------------------------------------------
" cscope database path 설정
"-------------------------------------------------------------------------------
    set csprg=/usr/local/bin/cscope                      "cscope 위치
    set csto=0                                           "cscope DB search first
    set cst                                              "cscope DB tag DB search
    set nocsverb                                         "verbose off
    "cscope DB의 위치설정, 절대경로 사용
    cs add /usr/src/linux-2.6.30.4/cscope.out /usr/src/linux-2.6.30.4
    set csverb                                           "verbose on
```

vim 환경 설정 (.vimrc) (2)

```
"-------------------------------------------------------------------------------
" Tag List 환경설정 2012.12.08
" nmap : normal mode key mapping
" Function Key가 매핑되게 하려면 환경변수 TERM이 xterm으로 설정돼야함
" putty에서 Terminal --> Keyboard --> Function Key 에서 Xterm R6로 설정함
"-------------------------------------------------------------------------------
    filetype on                                          "vim filetype on
    nmap <F7> :TlistToggle<CR>                           "F7 Key = Tag List Toggling
    let Tlist_Ctags_Cmd = "/usr/local/bin/ctags"         "ctags 프로그램 위치
    let Tlist_Inc_Winwidth = 0                           "window width change off
    let Tlist_Exit_OnlyWindow = 0                        "tag/file 선택 완료 시 taglist
                                                         "window close = off
    let Tlist_Auto_Open = 0                              "vim 시작 시 window open = off
    let Tlist_Use_Right_Window = 1                       "Tag List 위치 = 오른쪽
```

vim 환경 설정 (.vimrc) (3)

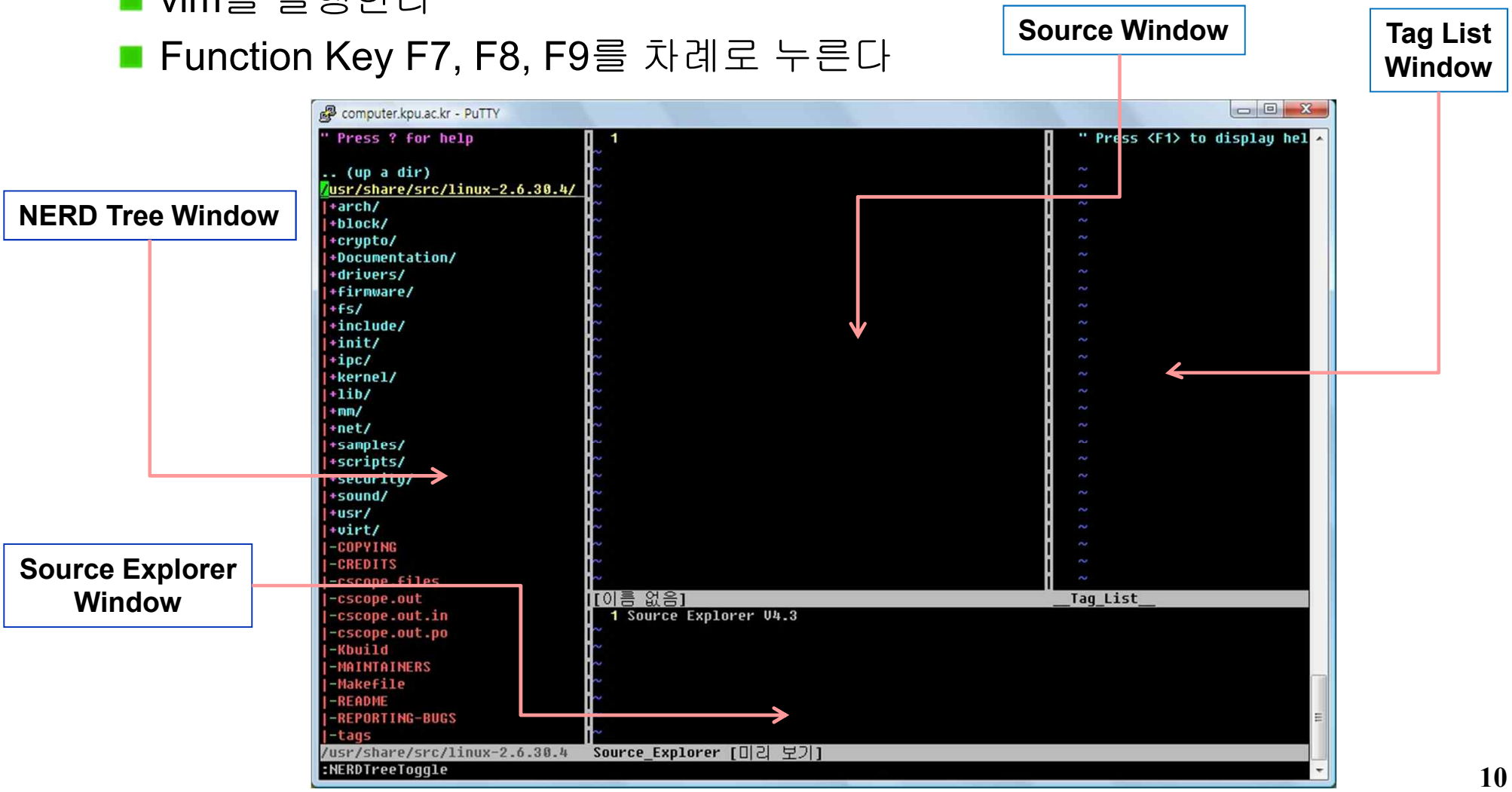
```
"------------------------------------------------------------------------------
" Source Explorer 환경설정
"------------------------------------------------------------------------------
    nmap <F8> :SrcExplToggle<CR>                    "F8 Key = SrcExpl Toggling
    nmap <C-H> <C-W>h                               "왼쪽 창으로 이동
    nmap <C-J> <C-W>j                               "하단(preview) 창으로 이동
    nmap <C-K> <C-W>k                               "상단  창으로 이동
    nmap <C-L> <C-W>l                               "오른쪽 창으로 이동

    let g:SrcExpl_winHeight = 8                     "SrcExpl 윈도우 높이 지정
    let g:SrcExpl_refreshTime = 100                 "refreshing time = 100ms
    let g:SrcExpl_jumpKey = "<ENTER>"               "해당 definition으로 jump
    let g:SrcExpl_gobackKey = "<SPACE>"             "back
    let g:SrcExpl_isUpdateTags = 0                  "tag file update = off

"------------------------------------------------------------------------------
" NERD Tree 환경설정
"------------------------------------------------------------------------------
    let NERDTreeWinPos = "left"                     "NERD Tree위치 = 왼쪽
    nmap <F9> :NERDTreeToggle<CR>                   "F9 Key = NERD Tree Toggling
```

소스 분석 환경 툴 둘러 보기 (1)

- 소스코드 디렉토리로 이동
  - `cd /usr/src/linux-2.6.30.4`
- vim을 실행한다
- Function Key F7, F8, F9를 차례로 누른다



소스 분석 환경 툴 둘러 보기 (2)

- NERD Tree에서 ./init 디렉토리로 이동
- main.c 파일로 커서를 이동하여 엔터키 입력
- 태그 리스트 창으로 이동 (Ctrl + L 키 입력)
- start_kernel 심볼를 찾아 엔터키 입력
- (소스 윈도우)소스코드 상에서 커서를 이동하면 하단 Source Explorer 창에 해당 심볼이 정의된 곳을 보여줌
- start_kernel 심볼에 커서를 위치하고 Ctrl + J를 눌러 하단 Source Explorer 창으로 이동하여 2번 파일 선택 후 엔터
- 직전 위치로 돌아가려면 스페이스 키를 입력

소스 분석 환경 툴 둘러 보기 (3)

- ctags 로 검색안되는 심볼에 대해 cscope를 사용하는 예
- NERD Tree 창에서 ./kernel/kthread.c 파일 선택하여 오픈
- 247번 라인의 kthread_create_list 심볼에 커서를 위치시키면 Source Explorer 창에 “Definition Not Found” 메시지가 나옴
- (소스 윈도우) ex 모드에서 심볼을 찾기위한 cscope 명령을 입력
  - :cs find s kthread_create_list
  - 1 입력 후 엔터

일반적인 **ctags, cscope** 명령어 사용 방법

- 태그 생성 방법
  - ctags –R (특정 디렉토리 아래 있는 소스 파일들에 대해 태그를 생성 할 때)

- cscope 데이터베이스 생성방법
  - find . \(-name '*.c' -o -name '*.cpp' -o -name '*.cc' -o -name '*.h' -o -name '*.s' -o -name '*.S' \) -print > cscope.files
  - cscope -i cscope.files
  - /bin/mkcscope.sh 명령어 참조

자주 사용하는 **ctags** 명령

명령	설명
Ctrl+]	함수가 정의된 곳으로 이동
Ctrl+t	이동하기 전 단계로 이동
:tselect <function-name>	function-name 목록 리스트
:ta keyword	keyword와 일치하는 태그 위치로 이동
:tn	검색 결과가 여러 개일 때 다음 결과로 이동 (:tnext)
:tp	이전 결과로 이동 (:tprevious)
:tfirst	첫 번째 목록으로 이동
:tlast	마지막 목록으로 이동
:sts <태그>	<태그>가 정의된 위치를 나열하고 선택한 위치로 창을 수평 분할해서 새로 생성된 창에 표시

자주 사용하는 **cscope** 명령 **(querytype)**

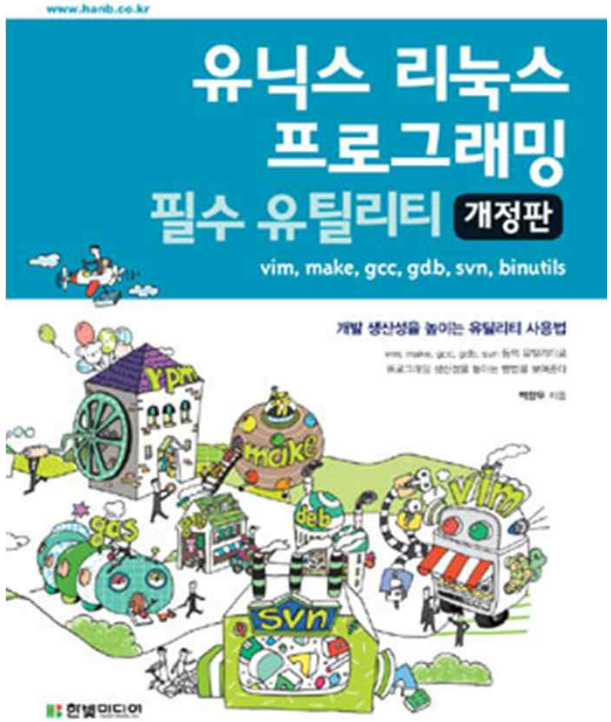
:cs find <querytype> <name>

<querytype>	설명
0 or s	C 심볼 찾기
1 or g	정의(definition) 찾기
2 or d	이 함수에 의해 호출되는 (called) 함수들 찾기
3 or c	이 함수를 호출하는 (calling) 함수들 찾기
4 or t	텍스트 스트링 (text string) 찾기
6 or e	egep 패턴 찾기
7 or f	파일 찾기
8 or i	이 파일을 #include로 포함하는 파일 찾기

cscope 도움말: **ex** 모드에서 ***:help cscope*** *또는* ***:cs help***

참고문헌

- 백창우, 유닉스 리눅스 프로그래밍 필수 유틸리: vim, make, gcc, gdb, svn, binutils (개정판), 한빛미디어, 2010



- 노서영 외, 코드로 알아보는 ARM 리눅스 커널, 제이펍, 2012