

Linux Kernel Source 분석 기법

- Kernel Source 분석 도구
 - ctags 설정 및 사용법
 - cscope 설정 및 사용법

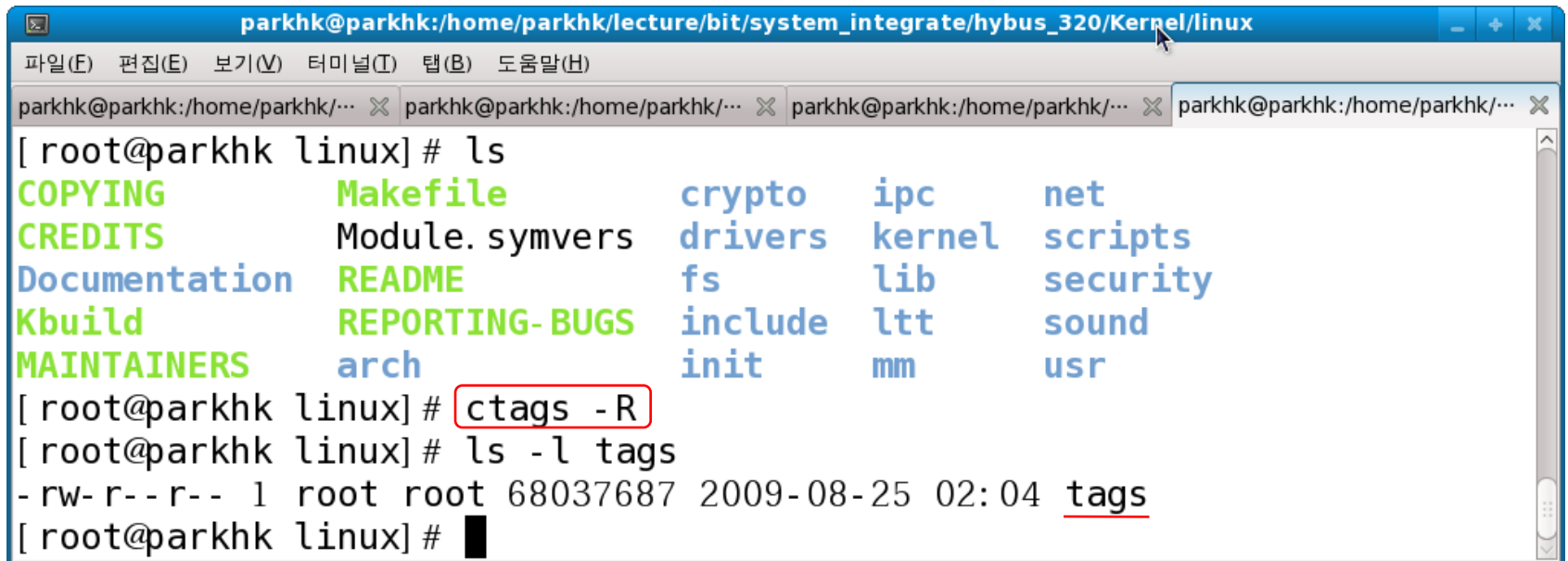
소스 분석 도구 ctags

- ctags의 정의
 - 소스분석을 위한 tag(기본 파일명 : tags) 파일을 생성하는 unix 명령어.
 - tag 파일은 vi[m] 또는 emacs 같은 편집기에서 symbol을 찾을 때 이용됨
- ctags에서 인식하는 파일의 종류

Assembler, ASP, Awk, C, C++, BETA, Cobol, Eiffel, Fortran, Perl, PHP, PL/SQL, Python	REXX, Ruby, Scheme, Shell scripts (Bourne/Korn/Z)	S-Lang, Tcl, Vera, Verilog, Vim, YACC/Bison
---	---	---

소스 분석 도구 ctags

- ctags 설정 및 기본 사용법
 - tag 파일 생성
 - 현재 디렉토리 이하 디렉토리에서 소스를 읽고 파싱하여 tag 파일을 생성



```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/parkhk/... x parkhk@parkhk:/home/parkhk/... x parkhk@parkhk:/home/parkhk/... x parkhk@parkhk:/home/parkhk/... x
[root@parkhk linux]# ls
COPYING          Makefile          crypto            ipc              net
CREDITS          Module.symvers    drivers           kernel           scripts
Documentation    README            fs               lib              security
Kbuild           REPORTING-BUGS    include          ltt              sound
MAINTAINERS      arch              init             mm               usr
[root@parkhk linux]# ctags -R
[root@parkhk linux]# ls -l tags
-rw-r--r-- 1 root root 68037687 2009-08-25 02:04 tags
[root@parkhk linux]# █
```

소스 분석 도구 ctags

- ctags 설정 및 기본 사용법
 - tags 파일

```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
start_irqno      include/asm-mips/jmr3927/irq.h  /^      int start_irqno;
$/"      m      struct: tb_irq_space
start_isoc_chain  drivers/isdn/hisax/hfc_usb.c    /^start_isoc_cha
in(usb_fifo * fifo, int num_packets_per_urb,$/;"      f      file:
start_ix         arch/cris/arch-v32/drivers/cryptocop.c /^      unsigned
int start_ix;$/;"      m      struct: cryptocop_tfrm_ctx      file:
start_join_net_params  drivers/net/wireless/rayctl.h  /^struct start_j
oin_net_params {$/;"      s
start_kcs_transaction  drivers/char/ipmi/ipmi_kcs_sm.c /^static int sta
rt_kcs_transaction(struct si_sm_data *kcs, unsigned char *data,$/;"
f      file:
start_kernel     arch/alpha/boot/bootp.c /^start_kernel(void)$/;"
start_kernel     arch/alpha/boot/bootpz.c /^start_kernel(void)$/;"
start_kernel     arch/alpha/boot/main.c /^void start_kernel(void)$/;"
start_kernel     init/main.c /^asmlinkage void __init start_kernel(vo
id)$/;" f
1 change; before #1 1 seconds ago 770577,1 92%
```

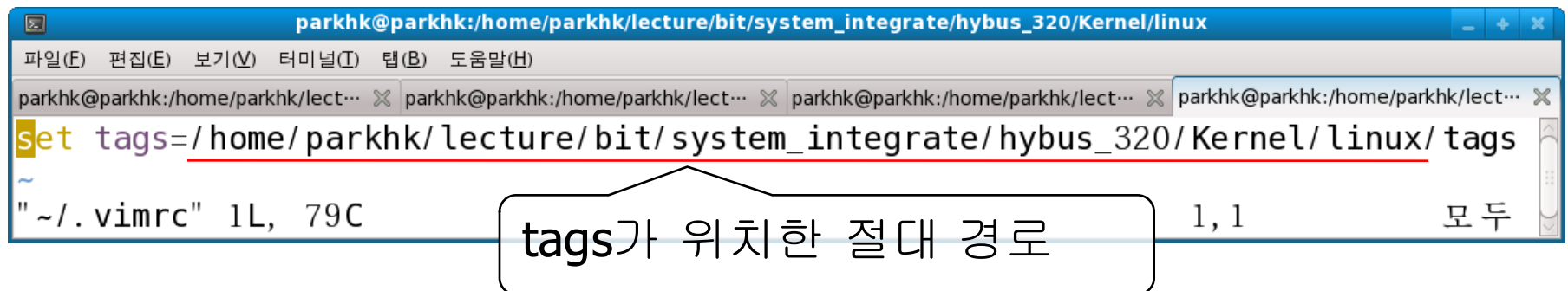
태그명

파일명

파일 내에 정의된 형식

소스 분석 도구 ctags

- tag 데이터 베이스 파일 설정(tags)
 - vim에서 set 명령어로 tags 변수에 생성한 tag file 설정
 - 홈 디렉토리 내에 .vimrc 파일에 다음 내용 기재
 - tags 파일이 복수개 일 경우 ;로 구분하고 작성



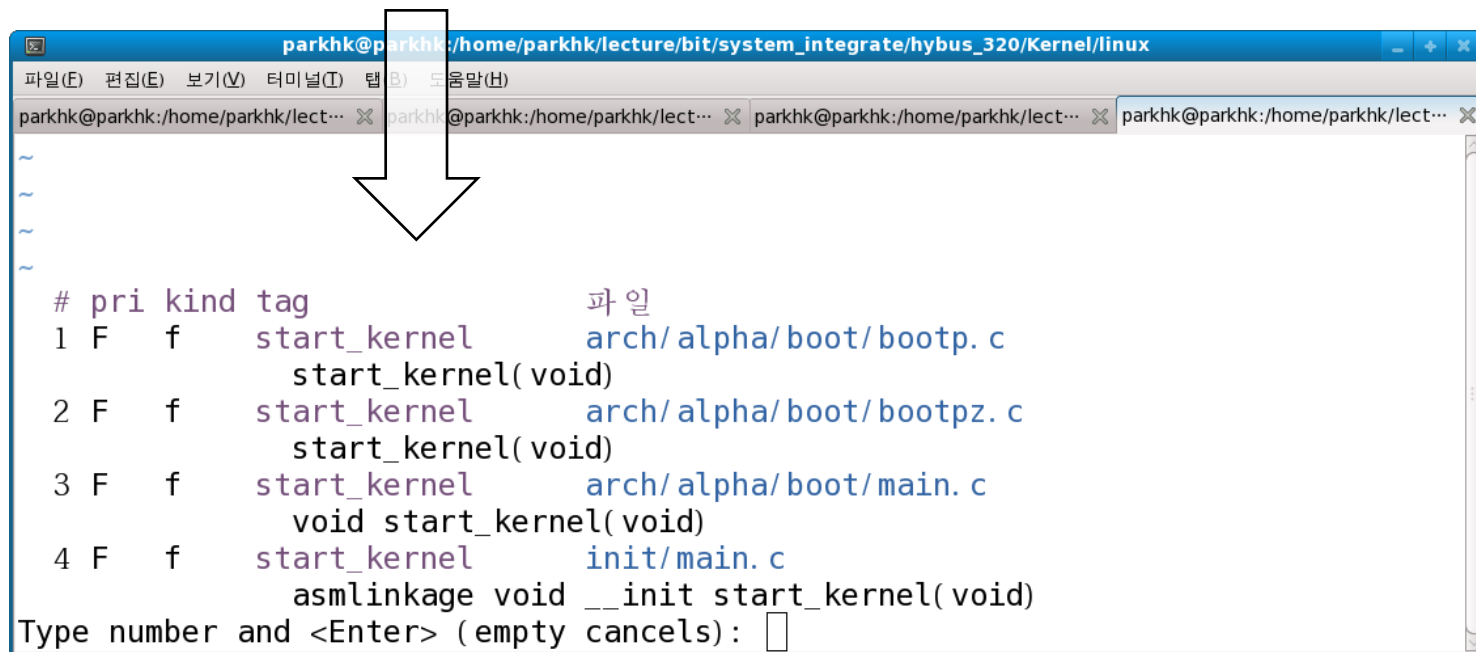
```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/parkhk/lect... x parkhk@parkhk:/home/parkhk/lect... x parkhk@parkhk:/home/parkhk/lect... x parkhk@parkhk:/home/parkhk/lect... x
set tags=/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux/tags
~
" ~/.vimrc" 1L, 79C tags가 위치한 절대 경로 1,1 모두
```

소스 분석 도구 ctags

- ctags의 태그 검색 방법
 1. 태그 찾는 명령 수행
 2. tags 파일의 태그명 필드와 비교하여 일치하는 태그
찾음
 3. 리스트를 출력함
 4. 사용자 선택에 따라 태그가 위치한 파일을 열고
커서를 태그에 위치 시킴

소스 분석 도구 ctags

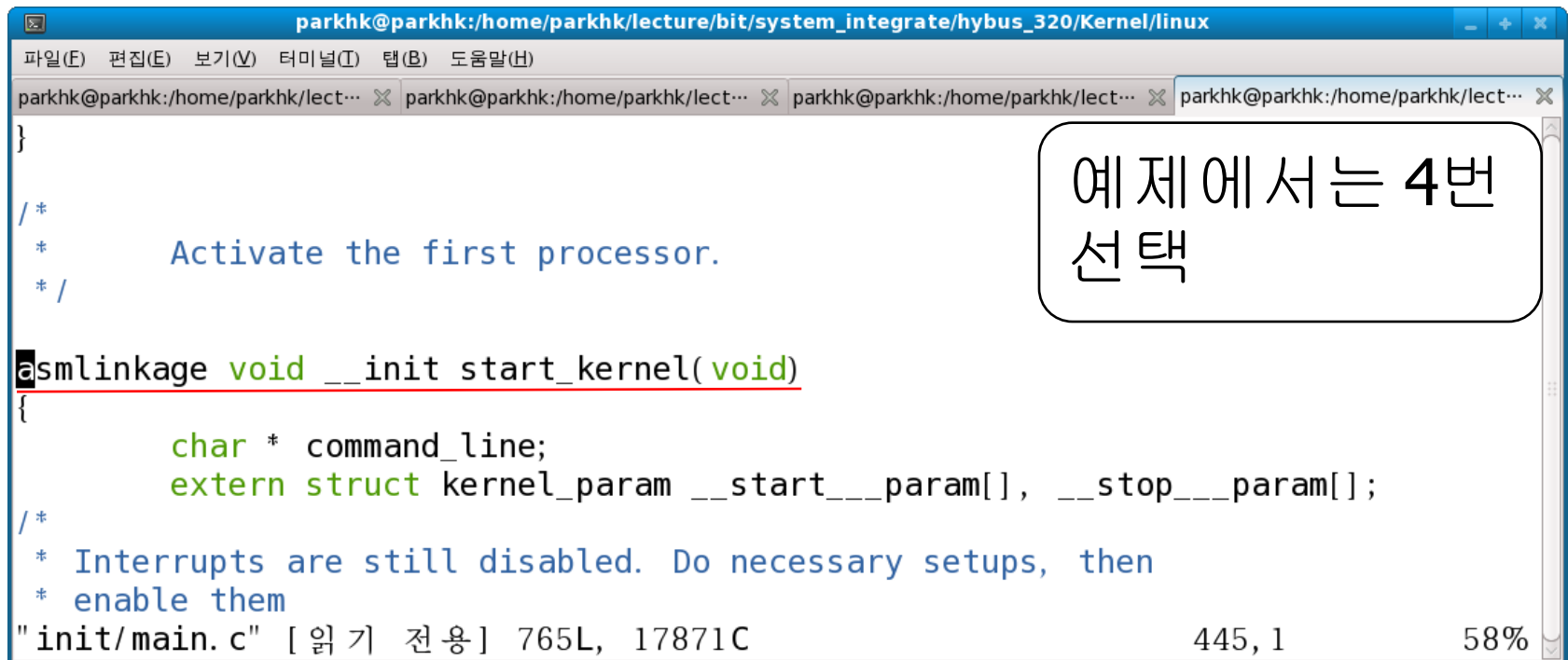
- ctags를 이용한 소스 분석
 - vi 수행
 - tj [찾는 태그]
 - ex) tj start_kernel



```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
~
~
~
# pri kind tag 파일
1 F f start_kernel arch/alpha/boot/bootp.c
start_kernel(void)
2 F f start_kernel arch/alpha/boot/bootpz.c
start_kernel(void)
3 F f start_kernel arch/alpha/boot/main.c
void start_kernel(void)
4 F f start_kernel init/main.c
asmlinkage void __init start_kernel(void)
Type number and <Enter> (empty cancels):
```


소스 분석 도구 ctags

- ctags를 이용한 소스 분석
 - 찾는 함수(또는 전역변수 등)에 대한 번호 선택

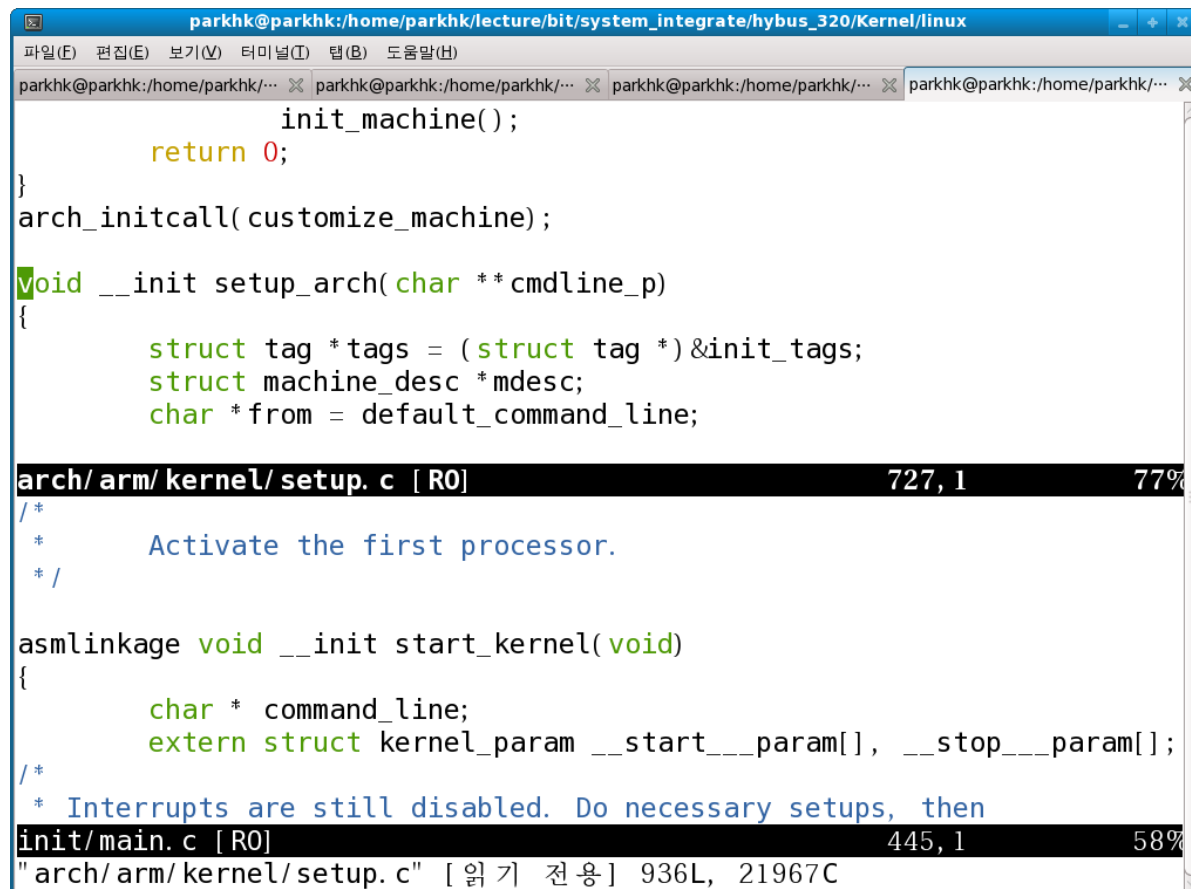


The screenshot shows a terminal window with a blue title bar. The window title is "parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux". The terminal content shows a C source code file with a function definition. A callout box on the right side of the terminal contains the text "예제에서는 4번 선택" (In the example, select 4 times).

```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/parkhk/lect... x parkhk@parkhk:/home/parkhk/lect... x parkhk@parkhk:/home/parkhk/lect... x parkhk@parkhk:/home/parkhk/lect... x
}
/*
 *   Activate the first processor.
 */
asm linkage void __init start_kernel(void)
{
    char * command_line;
    extern struct kernel_param __start__param[], __stop__param[];
/*
 * Interrupts are still disabled. Do necessary setups, then
 * enable them
"init/main.c" [읽기 전용] 765L, 17871C                                445, 1                                58%
```

소스 분석 도구 ctags

- ctags를 이용한 소스 분석
 - sts start_kernel



```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/parkhk/... parkhk@parkhk:/home/parkhk/... parkhk@parkhk:/home/parkhk/... parkhk@parkhk:/home/parkhk/...
    init_machine();
    return 0;
}
arch_initcall(customize_machine);
void __init setup_arch(char **cmdline_p)
{
    struct tag *tags = (struct tag *)&init_tags;
    struct machine_desc *mdesc;
    char *from = default_command_line;
arch/arm/kernel/setup.c [R0] 727, 1 77%
/*
 *   Activate the first processor.
 */
asmlinkage void __init start_kernel(void)
{
    char * command_line;
    extern struct kernel_param __start__param[], __stop__param[];
/*
 * Interrupts are still disabled. Do necessary setups, then
init/main.c [R0] 445, 1 58%
"arch/arm/kernel/setup.c" [읽기 전용] 936L, 21967C
```

소스 분석 도구 ctags

- ctag 명령어 요약

:ta [tag] or <CTRL> +]	{tag}가 정의된 위치를 나열하고 선택한 위치로 점프, 현재 위치는 stack에 push
:ts [tag] or :tj [tag]	
:po or <CTRL> + t	stack에 top에 저장된 위치를 pop하고 그 위치로 점프한다.
:sts [tag]	[tag]가 정의된 위치를 나열하고 선택한 위치로 창을 수평 분할하여 새로 생성된 창에 표시
:stj[tag]	
:tn	tj나ts로 점프했을 때 다음 tag로 점프
:tp	tj나ts로 점프했을 때 이전 tag로 점프
:tr	tj나ts로 점프했을 때 처음 tag로 점프
:tl	tj나ts로 점프했을 때 마지막 tag로 점프

소스 분석 도구 ctags

- ctag 명령어 요약

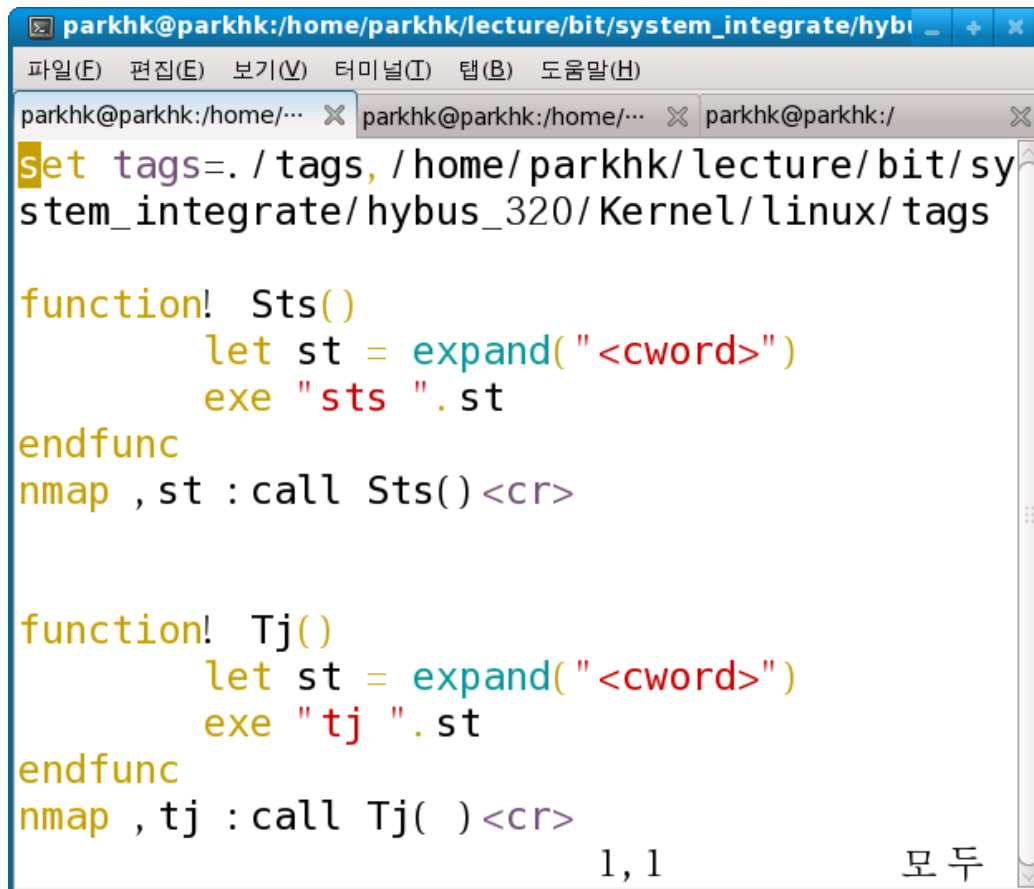
:pts[tag]	[tag]가 정의된 위치를 나열하고 선택한 위치로 창을 수평 분할하여 새로 생성된 창에 표시하지만 커서는 현재 위치 유지
:ptj[tag]	미리보기 윈도우에 tag가 정의된 형식을 보임
:ptn	pts나ptj로 점프했을 때 다음 tag로 점프
:ptp	pts나ptj로 점프했을 때 이전 tag로 점프
:ptr	pts나ptj로 점프했을 때 처음 tag로 점프
:ptl	pts나ptj로 점프했을 때 마지막 tag로 점프

소스 분석 도구 ctags

- 기타 사용법
 - `vi -t start_kernel`
 - vim 창에서 서한 것과 같음
 - 선택 창에 여러 개의 tag가 검색될 경우
 - space key : 페이지 넘기기
 - enter : 한 줄 넘기기
 - q : 선택을 위해 정지

소스 분석 도구 ctags

- Vim 내장 스크립트 작성
 - 홈 디렉토리의 .vimrc에 다음 내용 추가



```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybi
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/...  parkhk@parkhk:/home/...  parkhk@parkhk:/
set tags=., /tags, /home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux/tags

function! Sts()
    let st = expand("<word>")
    exe "sts ".st
endfunc
nmap ,st :call Sts()<cr>

function! Tj()
    let st = expand("<word>")
    exe "tj ".st
endfunc
nmap ,tj :call Tj( )<cr>

1,1 모두
```

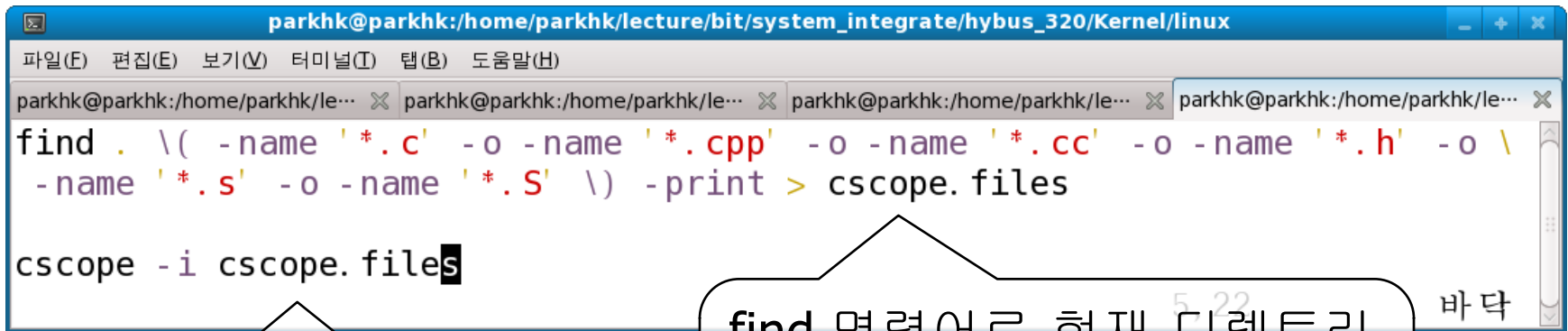
소스 분석 도구 cscope

- cscope?
 - ctags로 검색 할 수 없는 지역변수나 전역 변수, 그리고 함수가 사용된 곳을 검색
 - 심볼을 찾기 위한 데이터 베이스 파일로 cscope.out 파일을 사용
 - 즉, ctags 의 검색 기능을 강화하기 위해 사용
 - 분석 가능한 소스파일 목록

assembly, C, c++ ,Lex, Yacc등

소스 분석 도구 cscope

- cscope 설정 및 기본 사용법
 - cscope.out 파일 생성



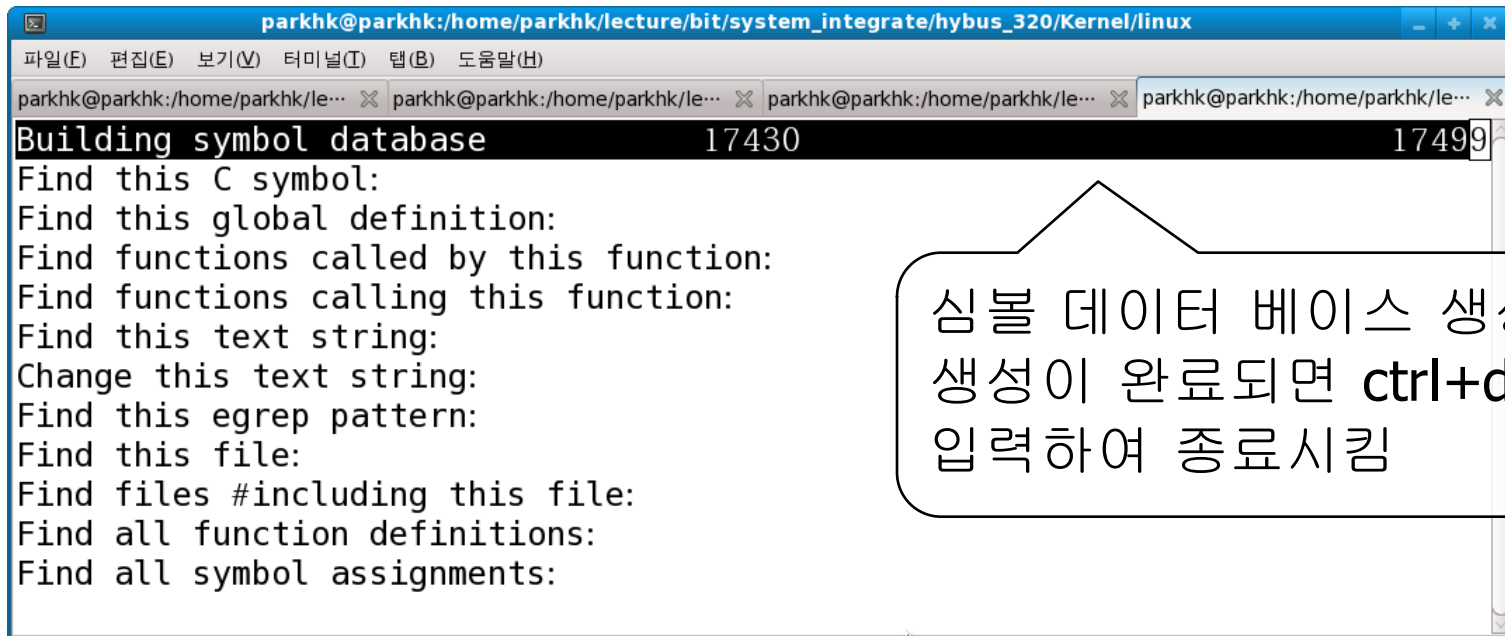
```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/parkhk/le... x parkhk@parkhk:/home/parkhk/le... x parkhk@parkhk:/home/parkhk/le... x parkhk@parkhk:/home/parkhk/le... x
find . \( -name '*.c' -o -name '*.cpp' -o -name '*.cc' -o -name '*.h' -o \
-name '*.s' -o -name '*.S' \) -print > cscope.files
cscope -i cscope.files
```

cscope가 한 줄씩
읽어들여서 cscope.out 심볼
데이터베이스 파일 생성

find 명령어로 현재 디렉토리
이하 모든 디렉토리에 있는
파일중 소스파일 검색

소스 분석 도구 cscope

- cscope 설정 및 기본 사용법
 - cscope.out 파일 생성

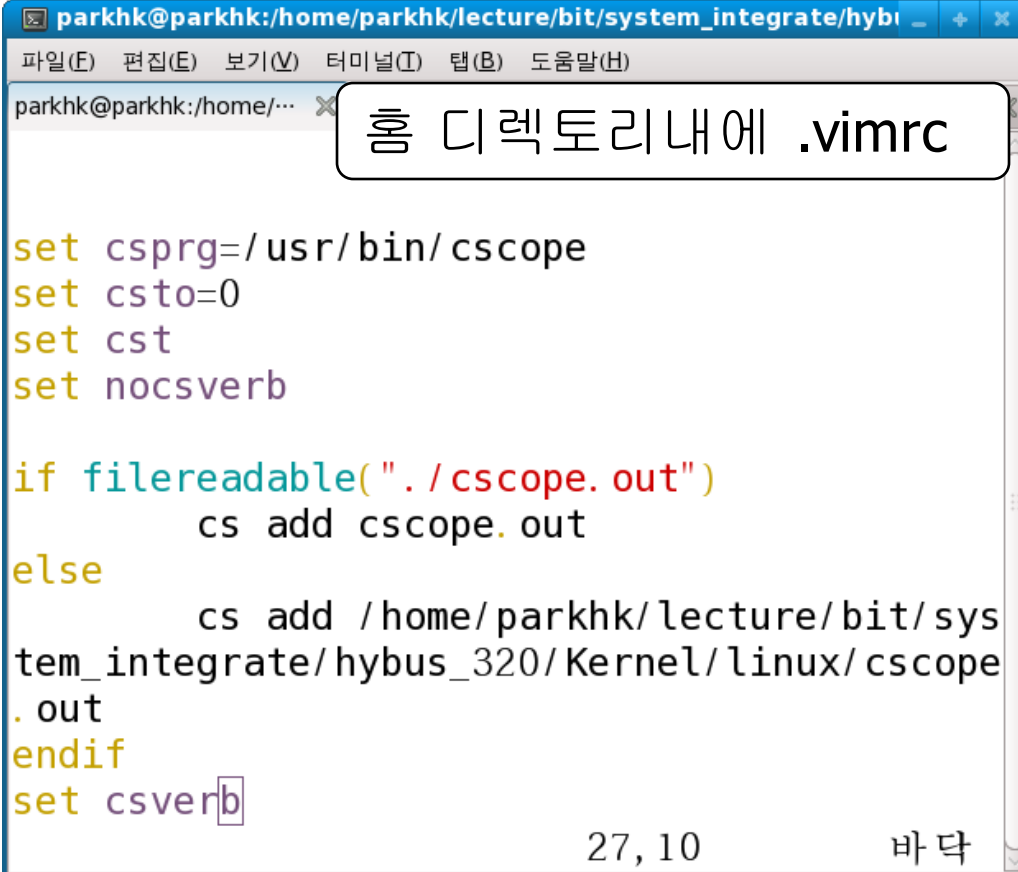


```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/parkhk/le... x parkhk@parkhk:/home/parkhk/le... x parkhk@parkhk:/home/parkhk/le... x parkhk@parkhk:/home/parkhk/le... x
Building symbol database 17430 17499
Find this C symbol:
Find this global definition:
Find functions called by this function:
Find functions calling this function:
Find this text string:
Change this text string:
Find this egrep pattern:
Find this file:
Find files #including this file:
Find all function definitions:
Find all symbol assignments:
```

심볼 데이터 베이스 생성
생성이 완료되면 **ctrl+d**를
입력하여 종료시킴

소스 분석 도구 cscope

- cscope 설정 및 기본 사용법
 - cscope 설정



The screenshot shows a terminal window with a blue title bar. The window title is "parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybi". The terminal content shows the configuration of the cscope tool in a .vimrc file. A callout box highlights the text "홈 디렉토리내에 .vimrc". The configuration includes setting the cscope program path, the output file name, and the directory to search for source files. The terminal shows the following code:

```
set csprg=/usr/bin/cscope
set csto=0
set cst
set nocsverb

if filereadable("./cscope.out")
    cs add cscope.out
else
    cs add /home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux/cscope.out
endif
set csverb
```

The terminal also shows the line number "27, 10" and the word "바닥" (bottom) at the bottom right.

소스 분석 도구 cscope

- cscope 설정 및 기본 사용법
 - cscope 사용법

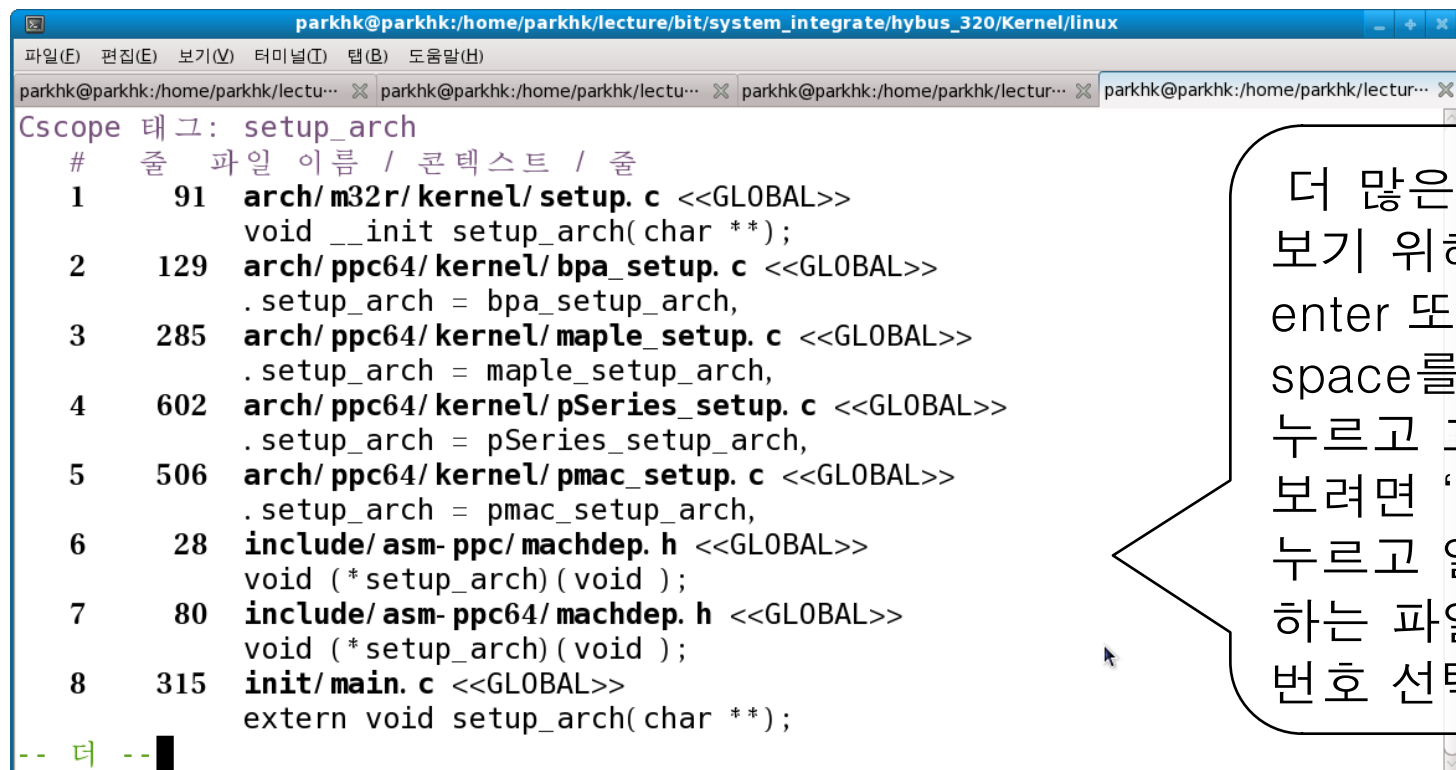
`:cs find [질의 종류] [심볼]`
ex) `:cs find s start_kernel`

질의
종류

0 or s	:	Find this C symbol
1 or g	:	Find this definition
2 or d	:	Find functions called by this function
3 or c	:	Find functions calling this function
4 or t	:	Find assignments to
6 or e	:	Find this egrep pattern
7 or f	:	Find this file
8 or i	:	Find files #including this file

소스 분석 도구 cscope

- cscope 설정 및 기본 사용법
 - 심볼 검색
 - :cs find s start_kernel



```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
Cscope 태그: setup_arch
# 줄 파일 이름 / 콘텍스트 / 줄
1 91 arch/m32r/kernel/setup.c <<GLOBAL>>
void __init setup_arch(char **);
2 129 arch/ppc64/kernel/bpa_setup.c <<GLOBAL>>
.setup_arch = bpa_setup_arch,
3 285 arch/ppc64/kernel/maple_setup.c <<GLOBAL>>
.setup_arch = maple_setup_arch,
4 602 arch/ppc64/kernel/pSeries_setup.c <<GLOBAL>>
.setup_arch = pSeries_setup_arch,
5 506 arch/ppc64/kernel/pmac_setup.c <<GLOBAL>>
.setup_arch = pmac_setup_arch,
6 28 include/asm-ppc/machdep.h <<GLOBAL>>
void (*setup_arch)(void);
7 80 include/asm-ppc64/machdep.h <<GLOBAL>>
void (*setup_arch)(void);
8 315 init/main.c <<GLOBAL>>
extern void setup_arch(char **);
-- 더 --
```

더 많은 list를
보기 위해서는
enter 또는
space를
누르고 그만
보려면 'q'를
누르고 열고자
하는 파일의
번호 선택.

소스 분석 도구 cscope

- Vim 내장 스크립트 작성
 - 홈 디렉토리의 .vimrc에 다음 내용 추가



```
parkhk@parkhk:/home/parkhk/lecture/bit/system_integrate/hybus_320/Kernel/linux
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
parkhk@parkhk:/home/parkhk/lecture/bit/syste...  parkhk@parkhk:/home/parkhk/lecture/bit/syste...  parkhk@parkhk:/
func! Css()
    let css = expand("<word>")
    new
    exe "cs find s ".css
endfunc
nmap ,css :call Css()<cr>

func! Csc()
    let csc = expand("<word>")
    new
    exe "cs find c ".csc
endfunc
nmap ,csc :call Csc()<cr>

func! Csd()
    let csd = expand("<word>")
    new
    exe "cs find d ".csd
endfunc
nmap ,csd :call Csd()<cr>

func! Csg()
    let csg = expand("<word>")
    new
    exe "cs find g ".csg
endfunc
nmap ,csg :call Csg()<cr>

~/.vimrc [+]  
28, 0-1  
65% ~/.vimrc [+]  
56, 0-1  
바닥
```