

# GDB, ctags, cscope and make

---

Youjip Won



# `gdb`

# GDB, the tool

---

- What is GDB?

- Default debugger for GNU software system
- GDB can trace execution of program and support other functionalities.

- How to compile file that can be debugged by gdb?

- `$ gcc -g gdb_test.c -o gdb_test`

- Run gdb

- `$ gdb [FILENAME]`

- How to pass arguments to program in gdb?

- `$ gdb [FILENAME]`
- `(gdb) set args [argument]`

# GDB - basic commands

---

- (gdb) `continue` or `c`
  - Continue execution of program
- (gdb) `step` or `s`
  - Run a line of code in program. If the code calls function, it enter to inside of function
- (gdb) `next` or `n`
  - Run a line of code in program. If the code calls function, it not enter to inside of function but execute next line.
- (gdb) `finish`
  - Continue execution of program until current function returns
- (gdb) `return value`
  - Stop the execution of current function, and use value as return value

# GDB - basic commands (Cont.)

---

- (gdb) `list 90`
  - Print source code at 90 line
- (gdb) `list badfunc`
  - Print the source code of specific function that you specified
- (gdb) `set listsize n`
  - Set the number of lines that is printed when the list command is executed
  - default is 10

# GDB - print

---

- When you want to examine some variables

- (gdb) `whatis [variable]` : Print the type of variable
- (gdb) `print [variable]` : Print the value of variable

- Print

- (gdb) `print a->member`
- (gdb) `print add(1,2)`
- (gdb) `print /x value`
- (gdb) `print p->state`
- (gdb) `print p->pid`
- (gdb) `print (*p)`
- You can specify the format of output as use x, u, o, c keyword

# GDB - break

---

- When you want to stop execution of program at position you want
  - (gdb) break 31
  - (gdb) break func
  - (gdb) break hello.c:main
  - (gdb) break utilc:300
  - (gdb) info break
  - (gdb) delete 1 : (If you not specify number, all breakpoints are deleted)
- Print the backtrace information in current function
  - (gdb) backtrace
  - (gdb) backtrace n

# gdb\_test.c

```
1.  #include <stdio.h>
2.
3.  void print() {
4.      printf("hello world!\n");
5.  }
6.
7.  int main() {
8.
9.      int i;
10.     for (i=0; i<10; i++) print();
11.     return 0;
12. }
```



# GDB(Cont.)

```
Reading symbols from gdb_test...done.
(gdb) break 7
Breakpoint 1 at 0x40053f: file gdb_test.c, line 7.
(gdb) break print
Breakpoint 2 at 0x40052a: file gdb_test.c, line 4.
(gdb) run
Starting program: /home/sundoo/gdb_test

Breakpoint 1, main () at gdb_test.c:10
warning: Source file is more recent than executable.
10      for (i=0; i < 10; i++) print();
(gdb) continue
Continuing.

Breakpoint 2, print () at gdb_test.c:4
4      printf("Hello World!\n");
(gdb) next
Hello World!
5      }
(gdb) continue
Continuing.

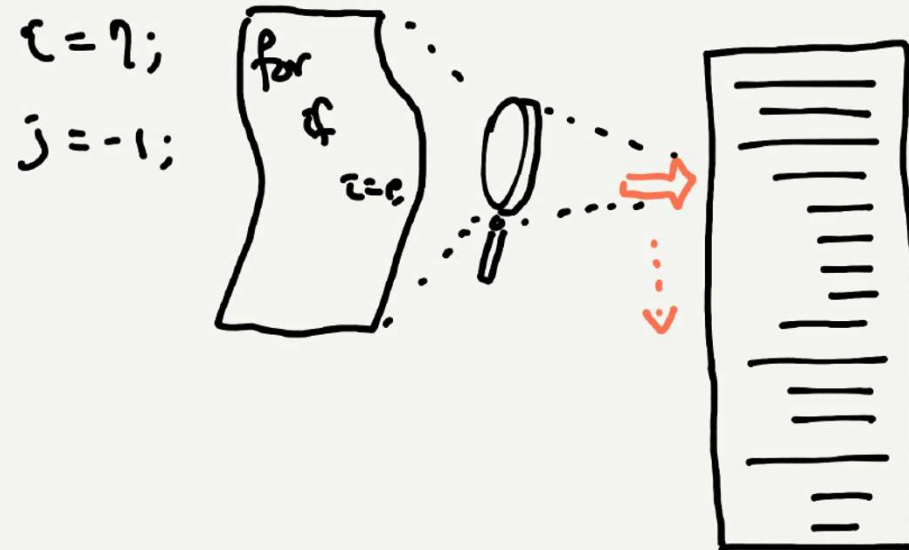
Breakpoint 2, print () at gdb_test.c:4
4      printf("Hello World!\n");
(gdb) bt
#0  print () at gdb_test.c:4
#1  0x000000000400552 in main () at gdb_test.c:10
(gdb) q
A debugging session is active.

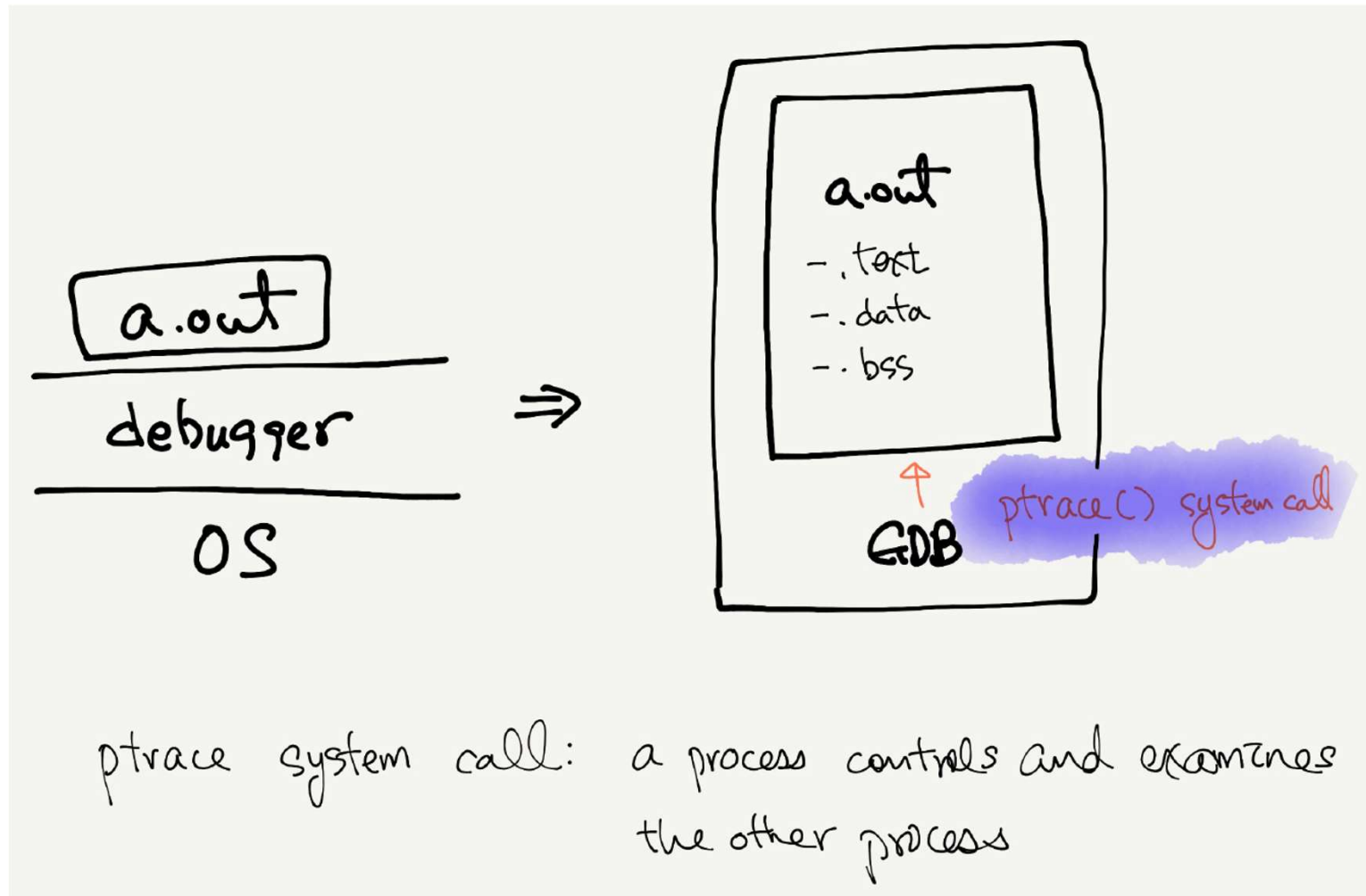
        Inferior 1 [process 20018] will be killed.

Quit anyway? (y or n) y
```

# Debugging

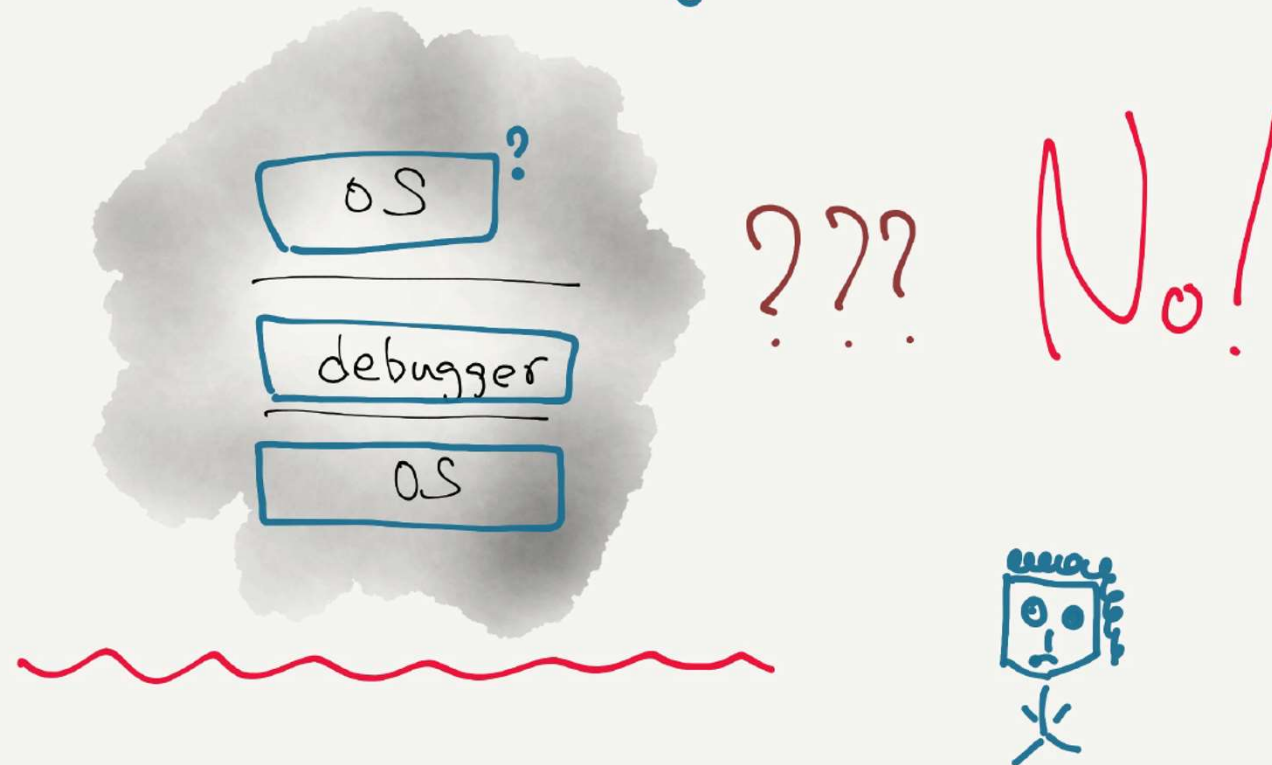
while the program is running,  
examine the program behavior.



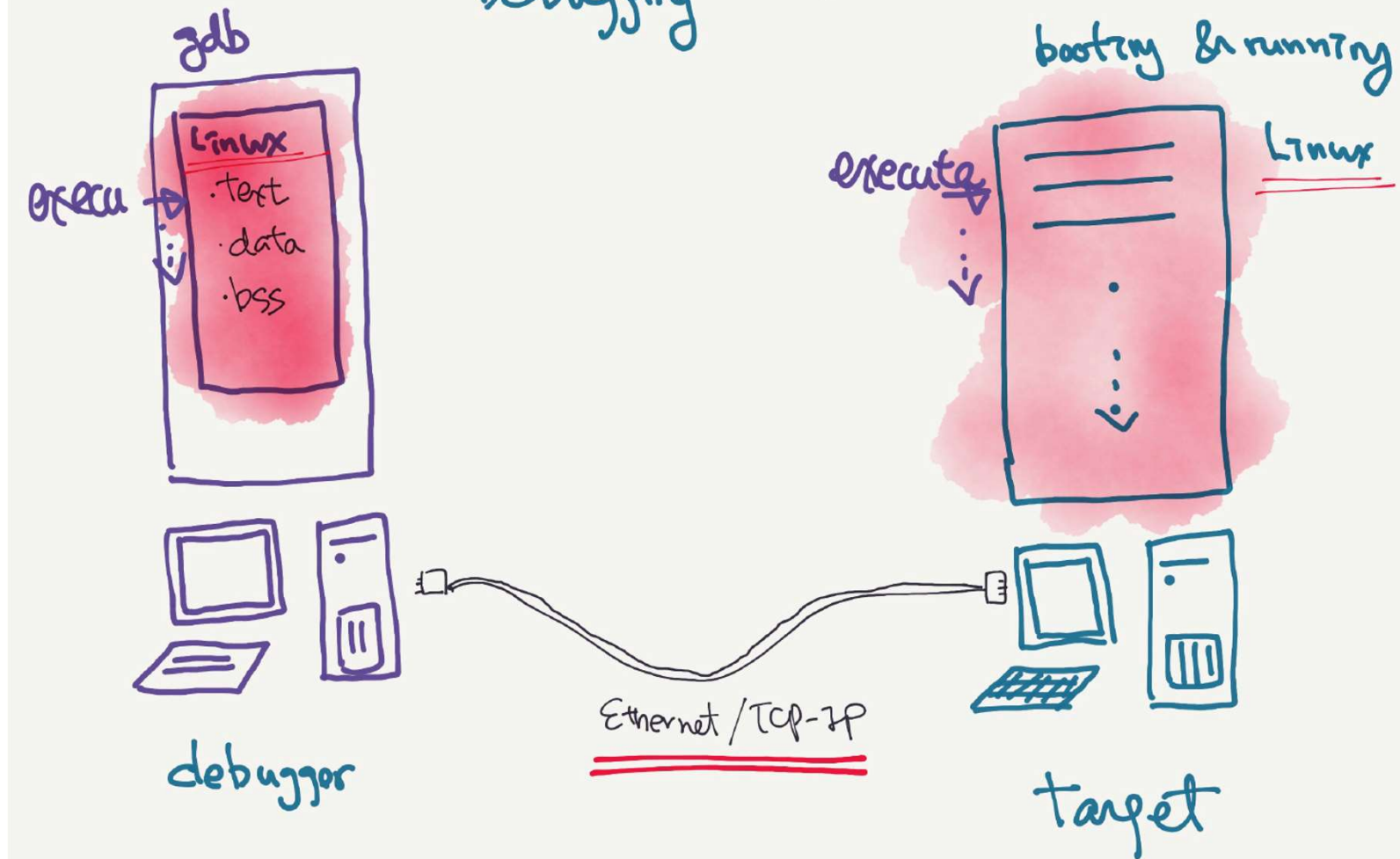


How to debug OS?

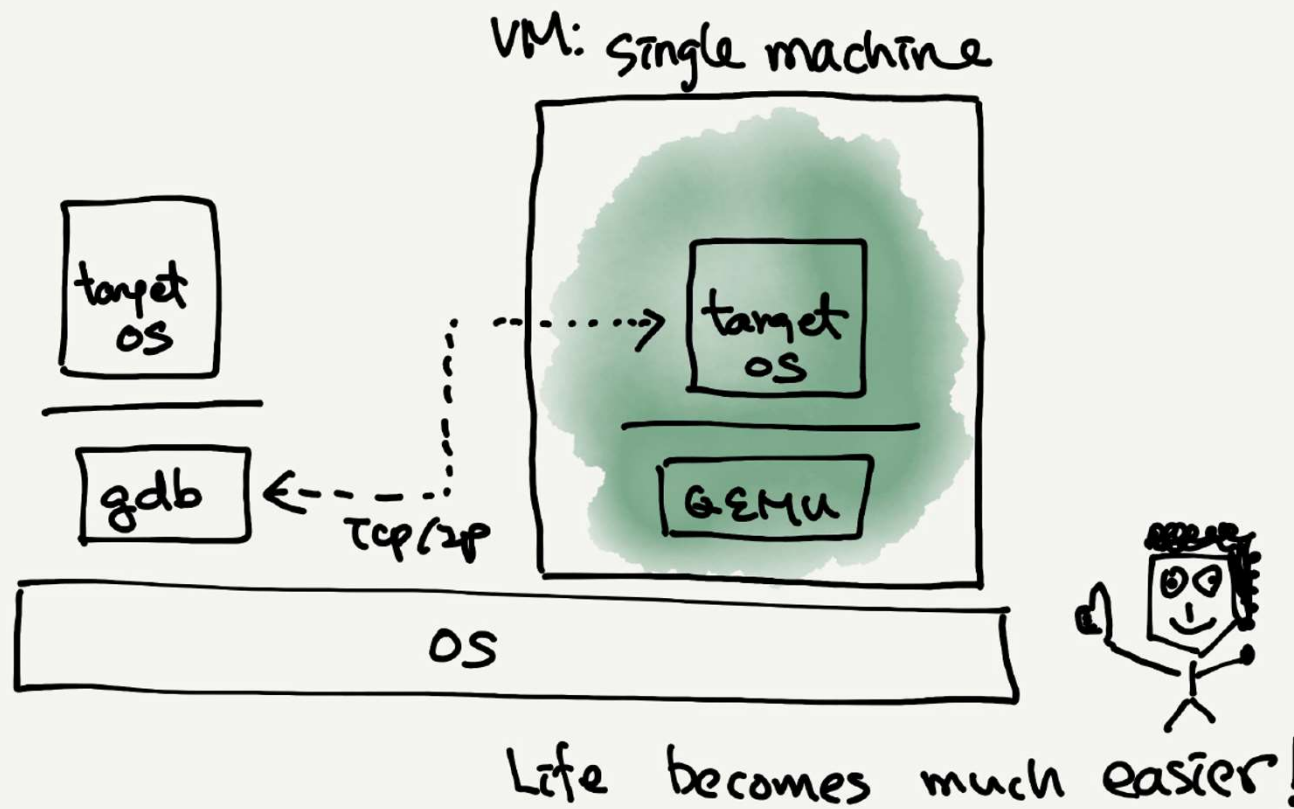
We need to debug OS!!!



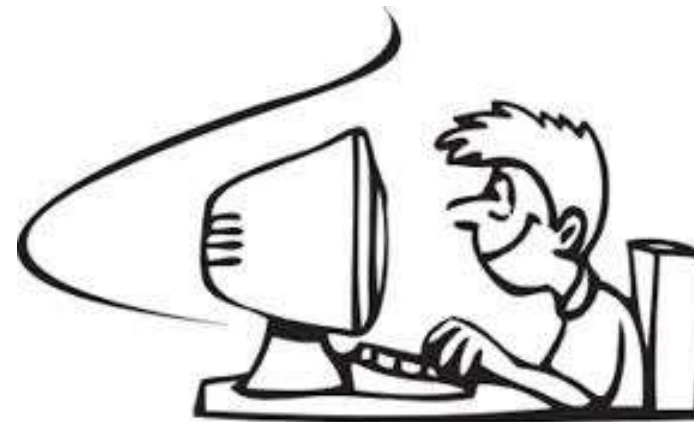
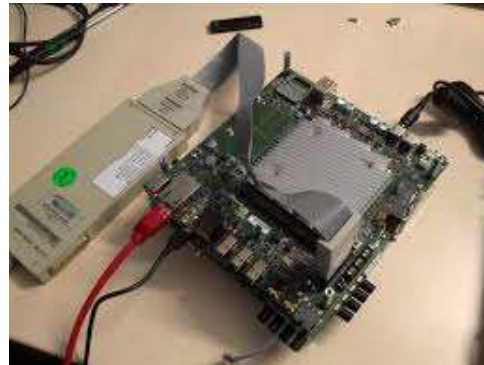
# Debugging OS



Debugging OS these days







# normal booting & debug mode booting

## • make qemu

```
• qemu-system-i386 -serial mon:stdio -drive  
  file=fs.img,index=1,media=disk,format=raw -drive  
  file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
```

## • make qemu-gdb

```
• qemu-system-i386 -serial mon:stdio -drive  
  file=fs.img,index=1,media=disk,format=raw -drive  
  file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512 -S  
-gdb tcp::25501
```

## • Difference between two upper command is “-S -gdb tcp::25501”

- -S : suspend the debug target just before the booting starts
- -gdb tcp::[port]
  - port number that is used to accept the command from gdb.



# First, run the debug target

```
$ qemu-system-i386 -serial mon:stdio -drive file=fs.img, index=1,media=disk,  
format=raw -drive file=xv6.img, index=0,media=disk,format=raw -smp 2 -m 512 -S  
-gdb tcp::25501
```

- execute upper command from the shell.
- QEMU stops and waits for the gdb command.



## Second, run the debugger

- Open a new terminal.
- Go to the directory where kernel binary resides.
- Execute gdb with loading the target binary.

```
$ gdb [binary file to load]
```

```
$ gdb kernel // linux & WSL
```

```
$ i386-elf-gdb kernel // macOS
```

```
dohyun@DESKTOP-LQ03DGV: ~/xv6-public$ gdb kernel
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from kernel...
+ target remote localhost:26000
The target architecture is assumed to be i8086
[f000:fff0] 0xffff0: jmp $0x3630,$0xf000e05b
0x0000fff0 in ?? ()
+ symbol-file kernel
(gdb)
```

Output of WSL

# .gdbinit

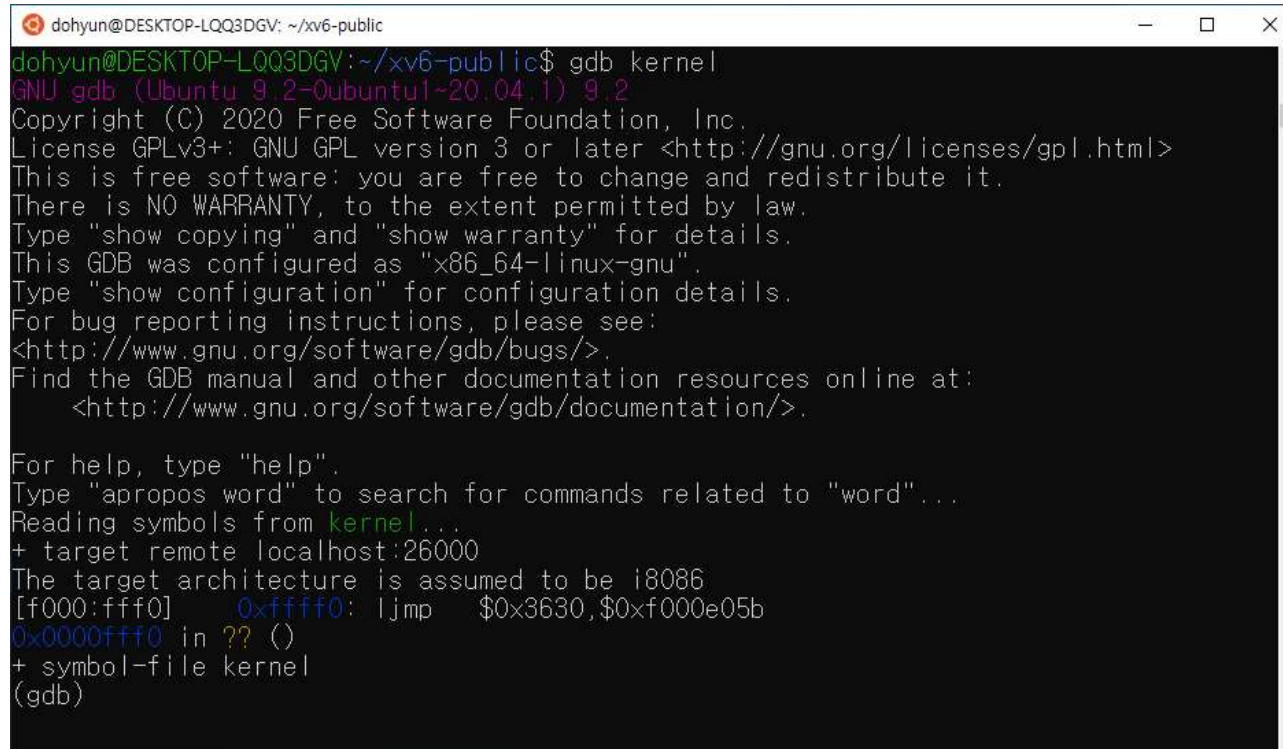
---

- GDB reads the `.gdbinit` first and executes the commands in this file.
- `.gdbinit` helps to automatically execute the commands you need to execute each time.

```
target remote localhost:26000
```

- Git repository of xv6-public already has `.gdbinit` file.
- If gdb is not connected to QEMU automatically, please check whether there is the `.gdbinit` file or not.

# connect the gdb to debug target



```
dohyun@DESKTOP-LQQ3DGV: ~/xv6-public
dohyun@DESKTOP-LQQ3DGV:~/xv6-public$ gdb kernel
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from kernel...
+ target remote localhost:26000
The target architecture is assumed to be i8086
[f000:fff0] 0xffff0: ljmp $0x3630,$0xf000e05b
0x0000fff0 in ?? ()
+ symbol-file kernel
(gdb)
```

- Load the kernel to gdb.
- It loads the program information and the symbols.
- It was not connected to the actual debug target.

# connect to the debug target

(gdb) target remote localhost:[port]

- Specify the port number with the one you

```
+ target remote localhost:26000
The target architecture is assumed to be i8086
[f000:fff0] 0xffff0: ljmp $0x3630,$0xf000e05b
0x0000fff0 in ?? ()
```

- setup the break point.

- (gdb) br main // set the breakpoint at main
- (gdb) c // continue

```
(gdb) br main
Breakpoint 1 at 0x80103040: file main.c, line 19.
(gdb) c
Continuing.
The target architecture is assumed to be i386
=> 0x80103040 <main>: endbr32

Thread 1 hit Breakpoint 1, main () at main.c:19
19 {
(gdb)
```

# connect to the debug target

(gdb) list

```
(gdb) list
14      // Bootstrap processor starts running C code here.
15      // Allocate a real stack and switch to it, first
16      // doing some setup required for memory allocator to work.
17      int
18      main(void)
19      {
20          kinit1(end, P2V(4*1024*1024)); // phys page allocator
21          kvmalloc();                  // kernel page table
22          mpinit();                    // detect other processors
23          lapicinit();                 // interrupt controller
(gdb) _
```

(gdb) br userinit

(gdb) c

(gdb) n or s

s: step into

n: next

# ctags

Materials from EE485 taught by  
Youjip Won and Kyungsoo Park

# ctags

---

- ctags
  - A tool for making “tag file” which contains the location of function, variable, string, and etc. of a source file.
  - You can use “tag file” to find the objects by name.
- ctags installation (linux)
  - `$ sudo apt-get install ctags`



# Setup ctags (1)

- Step1. building tag file

- \$ ctags [option] [filename(s)]

- example

- \$ ctags hello.c :making tag file which include index of “hello.c”

- \$ ctags -R :making the tag file for all subdirectories under current directory.

- option list

option	PATH
-R	Scan all subdirectory recursively.
--exclude=[pattern]	Exclude files and directories which have 'pattern' in name from creating tag file.
-x	Print tags as table to stdout without creating tag file.

# Setup ctags (2)

---

- Step2. Setting Tag file for Vim
  - open vim and type command below in vim command mode  
`set tags=[path of tag file]`
  - adding above command in ~/.vimrc will save this configuration

# Prepare the practice for ctags (1)

- Download the source code of bash with below command

```
$ git clone https://sourceware.org/git/glibc.git
```

- Move to the `glibc` directory in and Make `tags` file with below command

```
$ cd glibc  
$ ctags -R
```

- If you are using an old version of `ctags` so you cannot use the `-R` option, make `tags` file with below command (The eelab5 server has an old version)

```
$ cd glibc  
$ find . -name "*.[chS]" | ctags -
```

## Prepare the practice for the ctags (2)

- Open the `~/.vimrc` file

```
$ vi ~/.vimrc
```

- Type below text in `~/.vimrc` file

```
set tags+="Path of tha tags file you made"
```

```
set tags=~/.glibc/tags
```

```
"~/vimrc" 1 line --100%--
```

1,21

All

# Prepare the practice for the ctags (3)

- Save and quit the vi

```
:wq
```

- Open vi in glibc directory and run below command for testing

```
:tj fgets
```

```
~                               by Bram Moolenaar et al.
~                               Modified by pkg-vim-maintainers@lists.alioth.debian.org
~                               Vim is open source and freely distributable
~
~                               Help poor children in Uganda!
~                               type :help iccf<Enter>      for information
~
~                               type :q<Enter>             to exit
~                               type :help<Enter> or <F1>    for on-line help
~                               type :help version8<Enter>  for version info
~
~ # pri kind tag                file
1 F      fgets                 intl/localealias.c
~                               110
2 F      fgets                 intl/localealias.c
~                               # define fgets(buf, len, s) fgets_unlocked (buf, l
Type number and <Enter> (empty cancels): █
```

# ctags command in vim

Command	Description of the command
<code>:ta [tagname]</code>	Jump to definition of tagname
<code>:po</code>	Jump back to previous position
<code>Ctrl + ]</code>	Shortcut of command 'ta'. The tagname is a string pointed by current cursor.
<code>Ctrl + t</code>	Shortcut of command 'po'.
<code>:ts [tagname]</code>	Print list of definition of tagname
<code>:tj [tagname]</code>	Jump to definition of tagname if there is single definition Otherwise, print list of definition of tagname
<code>:tn</code>	Jump to next definition in list printed by command ts
<code>:tp</code>	Jump to previous definition in list printed by command ts
<code>:tags</code>	Print history of jump using ctags
<code>:sts</code>	Same with command 'ts', but split windows

[illegible]

**":ta fgets":**  
**Jump to definition of fgets**

```
1  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

:ta fgets█



```

258     __wur __warnattr ("fgets called with bigger size than length "
259                       "of destination buffer");
260
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)
263 {
264     if (__glibc_objsize (__s) != (size_t) -1)
265     {
266         if (!__builtin_constant_p (__n) || __n <= 0)
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);
268     }
269 }

```

"libio/bits/stdio2.h" 399L, 13353C 262,1 66%

# Shortcut of the command 'ta'

"Ctrl-]":

Jump to definition of function pointed by current cursor

```
258 __wur __warnattr ("fgets called with bigger size than length "  
259                  "of destination buffer");  
260  
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *  
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)  
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgetc_chk (__s, __glibc_objsize (__s), __n, __stream);  
267,10-17 66%
```



Type Ctrl-]

```
28 #include <stdio.h>  
29 #include <sys/param.h>  
30  
31 char *  
32 __fgetc_chk (char *buf, size_t size, int n, FILE *fp)  
33 {  
34     size_t count;  
35     char *result;  
36     CHECK_FILE (fp, NULL);  
37     if (n <= 0)  
"debug/fgetc_chk.c" 60L, 2234C 32,1 54%
```



# The command “po” and its shortcut

“:po” & “Ctrl-t”:

Jump to previous position (Continue to previous slide...)

```
28 #include <stdio.h>
29 #include <sys/param.h>
30
31 char *
32 __fgets_chk (char *buf, size_t size, int n, FILE *fp)
33 {
34     size_t count;
35     char *result;
36     CHECK_FILE (fp, NULL);
37     if (n <= 0)
:po
```



Type po or Ctrl-t

```
263 {
264     if (__glibc_objsize (__s) != (size_t) -1)
265     {
266         if (!_builtin_constant_p (__n) || __n <= 0)
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);
268
269         if ((size_t) __n > __glibc_objsize (__s))
270             return __fgets_chk_warn (__s, __glibc_objsize (__s), __n, __str
eam);
271     }
"libio/bits/stdio2.h" 399L, 13353C                                267,10-17    67%
```

# The command “ts”

“:ts fgets”:  
Print list of definition of fgets

```
258     __wur __warnattr ("fgets called with bigger size than length "  
259                        "of destination buffer");  
260  
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *  
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)  
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
:ts fgets
```



```
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
# pri kind tag          file  
1 F C f  fgets          libio/bits/stdio2.h  
          fgets (char *__restrict __s, int __n, FILE *__restrict __str  
          eam)  
2 FS  d  fgets          intl/localealias.c  
          110  
3 FS  d  fgets          intl/localealias.c  
          111  
Type number and <Enter> (empty cancels):
```

# The command “tj”

“:tj \_\_fgets\_chk”:  
Jump to definition of \_\_fgets\_chk  
\*It has single definition

```
258 __wur __warnattr ("fgets called with bigger size than length "  
259                  "of destination buffer");  
260  
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *  
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)  
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
:tj __fgets_chk
```



```
28 #include <stdio.h>  
29 #include <sys/param.h>  
30  
31 char *  
32 __fgets_chk (char *buf, size_t size, int n, FILE *fp)  
33 {  
34     size_t count;  
35     char *result;  
36     CHECK_FILE (fp, NULL);  
37     if (n <= 0)  
"debug/fgets_chk.c" 60L, 2234C          32,1          54%
```

“:tj fgets”:  
Print the list of definition of fgets  
\*It has multiple definitions

```
258 __wur __warnattr ("fgets called with bigger size than length "  
259                  "of destination buffer");  
260  
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *  
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)  
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
:tj fgets
```



```
266     if (!__builtin_constant_p (__n) || __n <= 0)  
267         return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
# pri kind tag      file  
1 F C f fgets      libio/bits/stdio2.h  
      fgets (char *__restrict __s, int __n, FILE *__restrict __str  
      eam)  
2 FS d fgets      intl/localealias.c  
      110  
3 FS d fgets      intl/localealias.c  
      111  
Type number and <Enter> (empty cancels):
```

Type the number of tag you want

# The command “tn” and “tp” (Cont. to “tj fgets”)

“:tn”:

Jump to next definition of fgets

```
258 __wur __warnattr ("fgets called with bigger size than length "  
259                  "of destination buffer");  
260  
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *  
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)  
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
:tn
```



```
106 # define freea(p) free (p)  
107 #endif  
108  
109 #if defined _LIBC_REENTRANT || defined HAVE_DECL_FGETS_UNLOCKED  
110 # undef fgets  
111 # define fgets(buf, len, s) fgets_unlocked (buf, len, s)  
112 #endif  
113 #if defined _LIBC_REENTRANT || defined HAVE_DECL_FEOF_UNLOCKED  
114 # undef feof  
115 # define feof(s) feof_unlocked (s)  
"intl/localealias.c" 431L, 10468C 110,1 24%
```

“:tp”:

Jump to previous definition of fgets

```
106 # define freea(p) free (p)  
107 #endif  
108  
109 #if defined _LIBC_REENTRANT || defined HAVE_DECL_FGETS_UNLOCKED  
110 # undef fgets  
111 # define fgets(buf, len, s) fgets_unlocked (buf, len, s)  
112 #endif  
113 #if defined _LIBC_REENTRANT || defined HAVE_DECL_FEOF_UNLOCKED  
114 # undef feof  
115 # define feof(s) feof_unlocked (s)  
:tp
```



```
258 __wur __warnattr ("fgets called with bigger size than length "  
259                  "of destination buffer");  
260  
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *  
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)  
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
"libio/bits/stdio2.h" 399L, 13353C 262,1 66%
```

# The command “tags” (Cont. to prev. slide)

“:tags”:  
Print history of jump using ctags

```
258     __wur __warnattr ("fgets called with bigger size than length "  
259                        "of destination buffer");  
260  
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *  
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)  
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
:tags
```



```
263 {  
264     if (__glibc_objsize (__s) != (size_t) -1)  
265     {  
266         if (!__builtin_constant_p (__n) || __n <= 0)  
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);  
:tags  
# TO tag          FROM line  in file/text  
1 1 fgets          1 /* Checking macros for stdio functions.  
2 1 fgets          262 fgets (char *__restrict __s, int __n, FILE *  
>  
Press ENTER or type command to continue
```



# The command “sts” (Cont. to prev. slide)

“:sts \_\_fgets\_chk”:  
Same with command ‘ts’, but split windows

```
250 __wur __attr_access ((__write_only__, 1, 2));
251 extern char *__REDIRECT (__fgets_alias,
252 (char *__restrict __s, int __n,
253 FILE *__restrict __stream), fgets)
254 __wur __attr_access ((__write_only__, 1, 2));
255 extern char *__REDIRECT (__fgets_chk_warn,
256 (char *__restrict __s, size_t __size, int __n,
257 FILE *__restrict __stream), __fgets_chk)
258 __wur __warnattr ("fgets called with bigger size than length "
259 "of destination buffer");
260
261 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *
262 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)
263 {
264     if (__glibc_objsize (__s) != (size_t) -1)
265     {
266         if (!__builtin_constant_p (__n) || __n <= 0)
267             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);
268
269         if ((size_t) __n > __glibc_objsize (__s))
270             return __fgets_chk_warn (__s, __glibc_objsize (__s), __n, __stream);
271     }
272     return __fgets_alias (__s, __n, __stream);
273 }
274
275 extern size_t __fread_chk (void *__restrict __ptr, size_t __ptrlen,
276 size_t __size, size_t __n,
277 FILE *__restrict __stream) __wur;
278 extern size_t __REDIRECT (__fread_alias,
:sts __fgets_chk
```



```
26
27 #include "libioP.h"
28 #include <stdio.h>
29 #include <sys/param.h>
30
31 char *
32 __fgets_chk (char *buf, size_t size, int n, FILE *fp)
33 {
34     size_t count;
35     char *result;
36     CHECK_FILE (fp, NULL);
37     if (n <= 0)
38         return NULL;
39     IO_acquire_lock (fp);
40
41     result = fgets (buf, size, fp);
42     if (result != NULL)
43         count = strlen (result);
44     else
45         count = 0;
46     if (count > size)
47         __wur __warnattr ("fgets called with bigger size than length "
48 "of destination buffer");
49
50     return result;
51 }
52
53 __fortify_function __wur __attr_access ((__write_only__, 1, 2)) char *
54 fgets (char *__restrict __s, int __n, FILE *__restrict __stream)
55 {
56     if (__glibc_objsize (__s) != (size_t) -1)
57     {
58         if (!__builtin_constant_p (__n) || __n <= 0)
59             return __fgets_chk (__s, __glibc_objsize (__s), __n, __stream);
60
61         if ((size_t) __n > __glibc_objsize (__s))
62             return __fgets_chk_warn (__s, __glibc_objsize (__s), __n, __stream);
63     }
64     return __fgets_alias (__s, __n, __stream);
65 }
66
67 #endif
68
69 #endif
70
71 #endif
72
73 #endif
74
75 #endif
76
77 #endif
78
79 #endif
80
81 #endif
82
83 #endif
84
85 #endif
86
87 #endif
88
89 #endif
90
91 #endif
92
93 #endif
94
95 #endif
96
97 #endif
98
99 #endif
100
101 #endif
102
103 #endif
104
105 #endif
106
107 #endif
108
109 #endif
110
111 #endif
112
113 #endif
114
115 #endif
116
117 #endif
118
119 #endif
120
121 #endif
122
123 #endif
124
125 #endif
126
127 #endif
128
129 #endif
130
131 #endif
132
133 #endif
134
135 #endif
136
137 #endif
138
139 #endif
140
141 #endif
142
143 #endif
144
145 #endif
146
147 #endif
148
149 #endif
150
151 #endif
152
153 #endif
154
155 #endif
156
157 #endif
158
159 #endif
160
161 #endif
162
163 #endif
164
165 #endif
166
167 #endif
168
169 #endif
170
171 #endif
172
173 #endif
174
175 #endif
176
177 #endif
178
179 #endif
180
181 #endif
182
183 #endif
184
185 #endif
186
187 #endif
188
189 #endif
190
191 #endif
192
193 #endif
194
195 #endif
196
197 #endif
198
199 #endif
200
201 #endif
202
203 #endif
204
205 #endif
206
207 #endif
208
209 #endif
210
211 #endif
212
213 #endif
214
215 #endif
216
217 #endif
218
219 #endif
220
221 #endif
222
223 #endif
224
225 #endif
226
227 #endif
228
229 #endif
230
231 #endif
232
233 #endif
234
235 #endif
236
237 #endif
238
239 #endif
240
241 #endif
242
243 #endif
244
245 #endif
246
247 #endif
248
249 #endif
250
251 #endif
252
253 #endif
254
255 #endif
256
257 #endif
258
259 #endif
260
261 #endif
262
263 #endif
264
265 #endif
266
267 #endif
268
269 #endif
270
271 #endif
272
273 #endif
274
275 #endif
276
277 #endif
278
279 #endif
280
281 #endif
282
283 #endif
284
285 #endif
286
287 #endif
288
289 #endif
290
291 #endif
292
293 #endif
294
295 #endif
296
297 #endif
298
299 #endif
300
301 #endif
302
303 #endif
304
305 #endif
306
307 #endif
308
309 #endif
310
311 #endif
312
313 #endif
314
315 #endif
316
317 #endif
318
319 #endif
320
321 #endif
322
323 #endif
324
325 #endif
326
327 #endif
328
329 #endif
330
331 #endif
332
333 #endif
334
335 #endif
336
337 #endif
338
339 #endif
340
341 #endif
342
343 #endif
344
345 #endif
346
347 #endif
348
349 #endif
350
351 #endif
352
353 #endif
354
355 #endif
356
357 #endif
358
359 #endif
360
361 #endif
362
363 #endif
364
365 #endif
366
367 #endif
368
369 #endif
370
371 #endif
372
373 #endif
374
375 #endif
376
377 #endif
378
379 #endif
380
381 #endif
382
383 #endif
384
385 #endif
386
387 #endif
388
389 #endif
390
391 #endif
392
393 #endif
394
395 #endif
396
397 #endif
398
399 #endif
400
401 #endif
402
403 #endif
404
405 #endif
406
407 #endif
408
409 #endif
410
411 #endif
412
413 #endif
414
415 #endif
416
417 #endif
418
419 #endif
420
421 #endif
422
423 #endif
424
425 #endif
426
427 #endif
428
429 #endif
430
431 #endif
432
433 #endif
434
435 #endif
436
437 #endif
438
439 #endif
440
441 #endif
442
443 #endif
444
445 #endif
446
447 #endif
448
449 #endif
450
451 #endif
452
453 #endif
454
455 #endif
456
457 #endif
458
459 #endif
460
461 #endif
462
463 #endif
464
465 #endif
466
467 #endif
468
469 #endif
470
471 #endif
472
473 #endif
474
475 #endif
476
477 #endif
478
479 #endif
480
481 #endif
482
483 #endif
484
485 #endif
486
487 #endif
488
489 #endif
490
491 #endif
492
493 #endif
494
495 #endif
496
497 #endif
498
499 #endif
500
501 #endif
502
503 #endif
504
505 #endif
506
507 #endif
508
509 #endif
510
511 #endif
512
513 #endif
514
515 #endif
516
517 #endif
518
519 #endif
520
521 #endif
522
523 #endif
524
525 #endif
526
527 #endif
528
529 #endif
530
531 #endif
532
533 #endif
534
535 #endif
536
537 #endif
538
539 #endif
540
541 #endif
542
543 #endif
544
545 #endif
546
547 #endif
548
549 #endif
550
551 #endif
552
553 #endif
554
555 #endif
556
557 #endif
558
559 #endif
560
561 #endif
562
563 #endif
564
565 #endif
566
567 #endif
568
569 #endif
570
571 #endif
572
573 #endif
574
575 #endif
576
577 #endif
578
579 #endif
580
581 #endif
582
583 #endif
584
585 #endif
586
587 #endif
588
589 #endif
590
591 #endif
592
593 #endif
594
595 #endif
596
597 #endif
598
599 #endif
600
601 #endif
602
603 #endif
604
605 #endif
606
607 #endif
608
609 #endif
610
611 #endif
612
613 #endif
614
615 #endif
616
617 #endif
618
619 #endif
620
621 #endif
622
623 #endif
624
625 #endif
626
627 #endif
628
629 #endif
630
631 #endif
632
633 #endif
634
635 #endif
636
637 #endif
638
639 #endif
640
641 #endif
642
643 #endif
644
645 #endif
646
647 #endif
648
649 #endif
650
651 #endif
652
653 #endif
654
655 #endif
656
657 #endif
658
659 #endif
660
661 #endif
662
663 #endif
664
665 #endif
666
667 #endif
668
669 #endif
670
671 #endif
672
673 #endif
674
675 #endif
676
677 #endif
678
679 #endif
680
681 #endif
682
683 #endif
684
685 #endif
686
687 #endif
688
689 #endif
690
691 #endif
692
693 #endif
694
695 #endif
696
697 #endif
698
699 #endif
700
701 #endif
702
703 #endif
704
705 #endif
706
707 #endif
708
709 #endif
710
711 #endif
712
713 #endif
714
715 #endif
716
717 #endif
718
719 #endif
720
721 #endif
722
723 #endif
724
725 #endif
726
727 #endif
728
729 #endif
730
731 #endif
732
733 #endif
734
735 #endif
736
737 #endif
738
739 #endif
740
741 #endif
742
743 #endif
744
745 #endif
746
747 #endif
748
749 #endif
750
751 #endif
752
753 #endif
754
755 #endif
756
757 #endif
758
759 #endif
760
761 #endif
762
763 #endif
764
765 #endif
766
767 #endif
768
769 #endif
770
771 #endif
772
773 #endif
774
775 #endif
776
777 #endif
778
779 #endif
780
781 #endif
782
783 #endif
784
785 #endif
786
787 #endif
788
789 #endif
790
791 #endif
792
793 #endif
794
795 #endif
796
797 #endif
798
799 #endif
800
801 #endif
802
803 #endif
804
805 #endif
806
807 #endif
808
809 #endif
810
811 #endif
812
813 #endif
814
815 #endif
816
817 #endif
818
819 #endif
820
821 #endif
822
823 #endif
824
825 #endif
826
827 #endif
828
829 #endif
830
831 #endif
832
833 #endif
834
835 #endif
836
837 #endif
838
839 #endif
840
841 #endif
842
843 #endif
844
845 #endif
846
847 #endif
848
849 #endif
850
851 #endif
852
853 #endif
854
855 #endif
856
857 #endif
858
859 #endif
860
861 #endif
862
863 #endif
864
865 #endif
866
867 #endif
868
869 #endif
870
871 #endif
872
873 #endif
874
875 #endif
876
877 #endif
878
879 #endif
880
881 #endif
882
883 #endif
884
885 #endif
886
887 #endif
888
889 #endif
890
891 #endif
892
893 #endif
894
895 #endif
896
897 #endif
898
899 #endif
900
901 #endif
902
903 #endif
904
905 #endif
906
907 #endif
908
909 #endif
910
911 #endif
912
913 #endif
914
915 #endif
916
917 #endif
918
919 #endif
920
921 #endif
922
923 #endif
924
925 #endif
926
927 #endif
928
929 #endif
930
931 #endif
932
933 #endif
934
935 #endif
936
937 #endif
938
939 #endif
940
941 #endif
942
943 #endif
944
945 #endif
946
947 #endif
948
949 #endif
950
951 #endif
952
953 #endif
954
955 #endif
956
957 #endif
958
959 #endif
960
961 #endif
962
963 #endif
964
965 #endif
966
967 #endif
968
969 #endif
970
971 #endif
972
973 #endif
974
975 #endif
976
977 #endif
978
979 #endif
980
981 #endif
982
983 #endif
984
985 #endif
986
987 #endif
988
989 #endif
990
991 #endif
992
993 #endif
994
995 #endif
996
997 #endif
998
999 #endif
1000
1001 #endif
1002
1003 #endif
1004
1005 #endif
1006
1007 #endif
1008
1009 #endif
1010
1011 #endif
1012
1013 #endif
1014
1015 #endif
1016
1017 #endif
1018
1019 #endif
1020
1021 #endif
1022
1023 #endif
1024
1025 #endif
1026
1027 #endif
1028
1029 #endif
1030
1031 #endif
1032
1033 #endif
1034
1035 #endif
1036
1037 #endif
1038
1039 #endif
1040
1041 #endif
1042
1043 #endif
1044
1045 #endif
1046
1047 #endif
1048
1049 #endif
1050
1051 #endif
1052
1053 #endif
1054
1055 #endif
1056
1057 #endif
1058
1059 #endif
1060
1061 #endif
1062
1063 #endif
1064
1065 #endif
1066
1067 #endif
1068
1069 #endif
1070
1071 #endif
1072
1073 #endif
1074
1075 #endif
1076
1077 #endif
1078
1079 #endif
1080
1081 #endif
1082
1083 #endif
1084
1085 #endif
1086
1087 #endif
1088
1089 #endif
1090
1091 #endif
1092
1093 #endif
1094
1095 #endif
1096
1097 #endif
1098
1099 #endif
1100
1101 #endif
1102
1103 #endif
1104
1105 #endif
1106
1107 #endif
1108
1109 #endif
1110
1111 #endif
1112
1113 #endif
1114
1115 #endif
1116
1117 #endif
1118
1119 #endif
1120
1121 #endif
1122
1123 #endif
1124
1125 #endif
1126
1127 #endif
1128
1129 #endif
1130
1131 #endif
1132
1133 #endif
1134
1135 #endif
1136
1137 #endif
1138
1139 #endif
1140
1141 #endif
1142
1143 #endif
1144
1145 #endif
1146
1147 #endif
1148
1149 #endif
1150
1151 #endif
1152
1153 #endif
1154
1155 #endif
1156
1157 #endif
1158
1159 #endif
1160
1161 #endif
1162
1163 #endif
1164
1165 #endif
1166
1167 #endif
1168
1169 #endif
1170
1171 #endif
1172
1173 #endif
1174
1175 #endif
1176
1177 #endif
1178
1179 #endif
1180
1181 #endif
1182
1183 #endif
1184
1185 #endif
1186
1187 #endif
1188
1189 #endif
1190
1191 #endif
1192
1193 #endif
1194
1195 #endif
1196
1197 #endif
1198
1199 #endif
1200
1201 #endif
1202
1203 #endif
1204
1205 #endif
1206
1207 #endif
1208
1209 #endif
1210
1211 #endif
1212
1213 #endif
1214
1215 #endif
1216
1217 #endif
1218
1219 #endif
1220
1221 #endif
1222
1223 #endif
1224
1225 #endif
1226
1227 #endif
1228
1229 #endif
1230
1231 #endif
1232
1233 #endif
1234
1235 #endif
1236
1237 #endif
1238
1239 #endif
1240
1241 #endif
1242
1243 #endif
1244
1245 #endif
1246
1247 #endif
1248
1249 #endif
1250
1251 #endif
1252
1253 #endif
1254
1255 #endif
1256
1257 #endif
1258
1259 #endif
1260
1261 #endif
1262
1263 #endif
1264
1265 #endif
1266
1267 #endif
1268
1269 #endif
1270
1271 #endif
1272
1273 #endif
1274
1275 #endif
1276
1277 #endif
1278
1279 #endif
1280
1281 #endif
1282
1283 #endif
1284
1285 #endif
1286
1287 #endif
1288
1289 #endif
1290
1291 #endif
1292
1293 #endif
1294
1295 #endif
1296
1297 #endif
1298
1299 #endif
1300
1301 #endif
1302
1303 #endif
1304
1305 #endif
1306
1307 #endif
1308
1309 #endif
1310
1311 #endif
1312
1313 #endif
1314
1315 #endif
1316
1317 #endif
1318
1319 #endif
1320
1321 #endif
1322
1323 #endif
1324
1325 #endif
1326
1327 #endif
1328
1329 #endif
1330
1331 #endif
1332
1333 #endif
1334
1335 #endif
1336
1337 #endif
1338
1339 #endif
1340
1341 #endif
1342
1343 #endif
1344
1345 #endif
1346
1347 #endif
1348
1349 #endif
1350
1351 #endif
1352
1353 #endif
1354
1355 #endif
1356
1357 #endif
1358
1359 #endif
1360
1361 #endif
1362
1363 #endif
1364
1365 #endif
1366
1367 #endif
1368
1369 #endif
1370
1371 #endif
1372
1373 #endif
1374
1375 #endif
1376
1377 #endif
1378
1379 #endif
1380
1381 #endif
1382
1383 #endif
1384
1385 #endif
1386
1387 #endif
1388
1389 #endif
1390
1391 #endif
1392
1393 #endif
1394
1395 #endif
1396
1397 #endif
1398
1399 #endif
1400
1401 #endif
1402
1403 #endif
1404
1405 #endif
1406
1407 #endif
1408
1409 #endif
1410
1411 #endif
1412
1413 #endif
1414
1415 #endif
1416
1417 #endif
1418
1419 #endif
1420
1421 #endif
1422
1423 #endif
1424
1425 #endif
1426
1427 #endif
1428
1429 #endif
1430
1431 #endif
1432
1433 #endif
1434
1435 #endif
1436
1437 #endif
1438
1439 #endif
1440
1441 #endif
1442
1443 #endif
1444
1445 #endif
1446
1447 #endif
1448
1449 #endif
1450
1451 #endif
1452
1453 #endif
1454
1455 #endif
1456
1457 #endif
1458
1459 #endif
1460
1461 #endif
1462
1463 #endif
1464
1465 #endif
1466
1467 #endif
1468
1469 #endif
1470
1471 #endif
1472
1473 #endif
1474
1475 #endif
1476
1477 #endif
1478
1479 #endif
1480
1481 #endif
1482
1483 #endif
1484
1485 #endif
1486
1487 #endif
1488
1489 #endif
1490
1491 #endif
1492
1493 #endif
1494
1495 #endif
1496
1497 #endif
1498
1499 #endif
1500
1501 #endif
1502
1503 #endif
1504
1505 #endif
1506
1507 #endif
1508
1509 #endif
1510
1511 #endif
1512
1513 #endif
1514
1515 #endif
1516
1517 #endif
1518
1519 #endif
1520
1521 #endif
1522
1523 #endif
1524
1525 #endif
1526
1527 #endif
1528
1529 #endif
1530
1531 #endif
1532
1533 #endif
1534
1535 #endif
1536
1537 #endif
1538
1539 #endif
1540
1541 #endif
1542
1543 #endif
1544
1545 #endif
1546
1547 #endif
1548
1549 #endif
1550
1551 #endif
1552
1553 #endif
1554
1555 #endif
1556
1557 #endif
1558
1559 #endif
1560
1561 #endif
1562
1563 #endif
1564
1565 #endif
1566
1567 #endif
1568
1569 #endif
1570
1571 #endif
1572
1573 #endif
1574
1575 #endif
1576
1577 #endif
1578
1579 #endif
1580
1581 #endif
1582
1583 #endif
1584
1585 #endif
1586
1587 #endif
1588
1589 #endif
1590
1591 #endif
1592
1593 #endif
1594
1595 #endif
1596
1597 #endif
1598
1599 #endif
1600
1601 #endif
1602
1603 #endif
1604
1605 #endif
1606
1607 #endif
1608
1609 #endif
1610
1611 #endif
1612
1613 #endif
1614
1615 #endif
1616
1617 #endif
1618
1619 #endif
1620
1621 #endif
1622
1623 #endif
1624
1625 #endif
1626
1627 #endif
1628
1629 #endif
1630
1631 #endif
1632
1633 #endif
1634
1635 #endif
1636
1637 #endif
1638
1639 #endif
1640
1641 #endif
1642
1643 #endif
1644
1645 #endif
1646
1647 #endif
1648
1649 #endif
1650
1651 #endif
1652
1653 #endif
1654
1655 #endif
1656
1657 #endif
1658
1659 #endif
1660
1661 #endif
1662
1663 #endif
1664
1665 #endif
1666
1667 #endif
1668
1669 #endif
1670
1671 #endif
1672
1673 #endif
1674
1675 #endif
1676
1677 #endif
1678
1679 #endif
1680
1681 #endif
1682
1683 #endif
1684
1685 #endif
1686
1687 #endif
1688
1689 #endif
1690
1691 #endif
1692
1693 #endif
1694
1695 #endif
1696
1697 #endif
1698
1699 #endif
1700
1701 #endif
1702
1703 #endif
1704
1705 #endif
1706
1707 #endif
1708
1709 #endif
1710
1711 #endif
1712
1713 #endif
1714
1715 #endif
1716
1717 #endif
1718
1719 #endif
1720
1721 #endif
1722
1723 #endif
1724
1725 #endif
1726
1727 #endif
1728
1729 #endif
1730
1731 #endif
1732
1733 #endif
1734
1735 #endif
1736
1737 #endif
1738
1739 #endif
1740
1741 #endif
1742
1743 #endif
1744
1745 #endif
1746
1747 #endif
1748
1749 #endif
1750
1751 #endif
1752
1753 #endif
1754
1755 #endif
1756
1757 #endif
1758
1759 #endif
1760
1761 #endif
1762
1763 #endif
1764
1765 #endif
1766
1767 #endif
1768
1769 #endif
1770
1771 #endif
1772
1773 #endif
1774
1775 #endif
1776
1777 #endif
1778
1779 #endif
1780
1781 #endif
1782
1783 #endif
1784
1785 #endif
1786
1787 #endif
1788
1789 #endif
1790
1791 #endif
1792
1793 #endif
1794
1795 #endif
1796
1797 #endif
1798
1799 #endif
1800
1801 #endif
1802
1803 #endif
1804
1805 #endif
1806
1807 #endif
1808
1809 #endif
1810
1811 #endif
1812
1813 #endif
1814
1815 #endif
1816
1817 #endif
1818
1819 #endif
1820
1821 #endif
1822
1823 #endif
1824
1825 #endif
1826
1827 #endif
1828
1829 #endif
1830
1831 #endif
1832
1833 #endif
1834
1835 #endif
1836
1837 #endif
1838
1839 #endif
1840
1841 #endif
1842
1843 #endif
1844
1845 #endif
1846
1847 #endif
1848
1849 #endif
1850
1851 #endif
1852
1853 #endif
1854
1855 #endif
1856
1857 #endif
1858
1859 #endif
1860
1861 #endif
1862
1863 #endif
1864
1865 #endif
1866
1867 #endif
1868
1869 #endif
1870
1871 #endif
1872
1873 #endif
1874
1875 #endif
1876
1877 #endif
1878
1879 #endif
1880
1881 #endif
1882
1883 #endif
1884
1885 #endif
1886
1887 #endif
1888
1889 #endif
1890
1891 #endif
1892
1893 #endif
1894
1895 #endif
1896
1897 #endif
1898
1899 #endif
1900
1901 #endif
1902
1903 #endif
1904
1905 #endif
1906
1907 #endif
1908
1909 #endif
1910
1911 #endif
1912
1913 #endif
1914
1915 #endif
1916
1917 #endif
1918
1919 #endif
1920
1921 #endif
1922
1923 #endif
1924
1925 #endif
1926
1927 #endif
1928
1929 #endif
1930
1931 #endif
1932
1933 #endif
1934
1935 #endif
1936
1937 #endif
1938
1939 #endif
1940
1941 #endif
1942
1943 #endif
1944
1945 #endif
1946
1947 #endif
1948
1949 #endif
1950
1951 #endif
1952
1953 #endif
1954
1955 #endif
1956
1957 #endif
1958
1959 #endif
1960
1961 #endif
1962
1963 #endif
1964
1965 #endif
1966
1967 #endif
1968
1969 #endif
1970
1971 #endif
1972
1973 #endif
1974
1975 #endif
1976
1977 #endif
1978
1979 #endif
1980
1981 #endif
1982
1983 #endif
1984
1985 #endif
1986
1987 #endif
1988
1989 #endif
1990
1991 #endif
1992
1993 #endif
1994
1995 #endif
1996
1997 #endif
1998
1999 #endif
2000
2001 #endif
2002
2003 #endif
2004
2005 #endif
2006
2007 #endif
2008
2009 #endif
2010
2011 #endif
2012
2013 #endif
2014
2015 #endif
2016
2017 #endif
2018
2019 #endif
2020
2021 #endif
2022
2023 #endif
2024
2025 #endif
2026
2027 #endif
2028
2029 #endif
2030
2031 #endif
2032
2033 #endif
2034
2035 #endif
2036
2037 #endif
2038
2039 #endif
2040
2041 #endif
2042
2043 #endif
2044
2045 #endif
2046
2047 #endif
2048
2049 #endif
2050
2051 #endif
2052
2053 #endif
2054
2055 #endif
2056
2057 #endif
2058
2059 #endif
2060
2061 #endif
2062
2063 #endif
2064
2065 #endif
2066
2067 #endif
2068
2069 #endif
2070
2071 #endif
2072
2073 #endif
2074
2075 #endif
2076
2077 #endif
2078
2079 #endif
2080
2081 #endif
2082
2083 #endif
2084
2085 #endif
2086
2087 #endif
2088
2089 #endif
2090
2091 #endif
2092
2093 #endif
2094
2095 #endif
2096
2097 #endif
2098
2099 #endif
2100
2101 #endif
2102
2103 #endif
2104
2105 #endif

```

# cscope

Materials from EE485 taught by  
Youjip Won and Kyungsoo Park

# cscope

---

- The tool supports functionalities that are not in ctags.
- The functionalities include:
  - Find the definition of symbol.
  - Find the positions where a given function is called or a variable is accessed.
  - Find the functions that specific function calls.
  - Find text string.
- Installation (Linux)
  - `$ sudo apt-get install cscope`



# Setup cscope (1)

- Step 1. Make a file “cscope.files” containing list of files what you want to analyze in the `glibc` directory.

```
$ find . -name "*.chS" > cscope.files
```

- Result:

```
20205219@eelab5:~/glibc$ find ./ -name "*.chS" > cscope.files
20205219@eelab5:~/glibc$ head -n 10 cscope.files
./resource/ulimit.c
./resource/getrlimit.c
./resource/setrlimit.c
./resource/vlimit.c
./resource/setrlimit64.c
./resource/getpriority.c
./resource/getrusage.c
./resource/vtimes.c
./resource/nice.c
./resource/setpriority.c
20205219@eelab5:~/glibc$
```

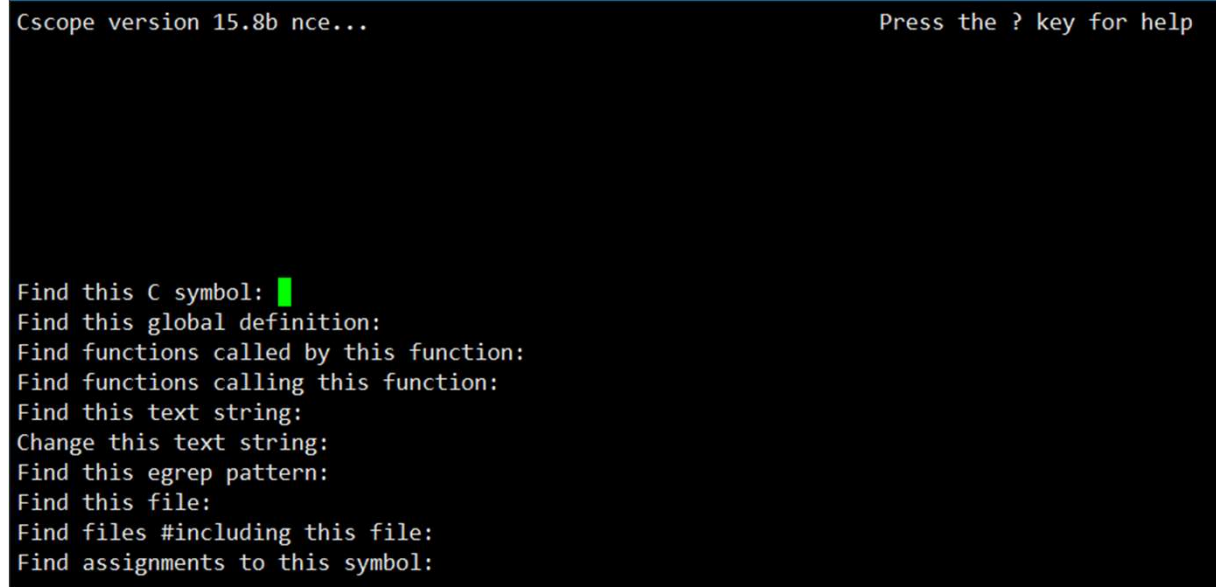
## Setup cscope (2)

- Step 2. Make a file “cscope.out” which is a database file for cscope with below command

```
$ cscope -i cscope.files
```

The cscope.out has the information of each file in cscope.files

- Result:



Cscope version 15.8b nce... Press the ? key for help

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 assignments to this symbol:

**\* You can leave the cscope with Ctrl-D**

## Setup cscope (3)

- Step 3. Set the cscope in vim with the command below.

```
: cs add ./cscope.out
```

- 🟡 Add above command in `~/.vimrc` will save this configuration.

[illegible]

# command

- Find the functions that “call” a given function or that “is called” by a given function.
  - Find all functions that call malloc()
  - Find all functions that are called by malloc().

**cs find** <Command Character> <String>

Command	Description
s	Find C symbol <i>String</i>
g	Find definition <i>String</i>
d	Find functions called by function <i>String</i>
c	Find functions calling function <i>String</i>
t	Find text <i>String</i>
e	Find egrep pattern <i>String</i>
f	Find file <i>String</i>
i	Find files #including file <i>String</i>

# Usage of cscope in vim

"cs find s printf":  
Search the symbol "printf"

```
Cscope tag: printf
# line filename / context / line
1 130 elf/tst-auditmod1.c <<GLOBAL>>
    printf ("pltexit: symname=%s, st_value=%#lx, ndx=%u, retval=%tu\n",
2 39 elf/tst-dlsym-error.c <<GLOBAL>>
    printf ("error: asprintf: %m\n");
3 46 elf/tst-dlsym-error.c <<GLOBAL>>
    printf ("error: %s: found symbol \"no_such_symbol\"\n", name);
4 52 elf/tst-dlsym-error.c <<GLOBAL>>
    printf ("error: %s: missing error message\n", name);
5 58 elf/tst-dlsym-error.c <<GLOBAL>>
    printf ("error: %s: unexpected error message: %s\n", name, message);
6 66 elf/tst-dlsym-error.c <<GLOBAL>>
    printf ("error: %s: unexpected error message: %s\n", name, message);
7 81 elf/tst-dlsym-error.c <<GLOBAL>>
    printf ("error: cannot dlopen %s: %s\n", LIBC_SO, dlerror ());
8 115 libio/bits/stdio2.h <<GLOBAL>>
    #define printf(...) \
-- More --
```

# Usage of cscope in vim

"cs find e ..printf":  
Search the symbol ..printf with regular expression

```
Cscope tag: ..printf
# line filename / context / line
1 103 argp/argp-fmtstream.c <<<unknown>>>
    __fxprintf (fs->stream, "%.*s", (int) (fs->p - fs->buf), fs->buf);
2 292 argp/argp-fmtstream.c <<<unknown>>>
    __fxprintf (fs->stream, "%.*s\n",
3 359 argp/argp-fmtstream.c <<<unknown>>>
    __fxprintf (fs->stream, "%.*s", (int) (fs->p - fs->buf), fs->buf);
4 400 argp/argp-fmtstream.c <<<unknown>>>
    __argp_fmtstream_printf (struct argp_fmtstream *fs, const char *fmt, ...)
5 415 argp/argp-fmtstream.c <<<unknown>>>
    out = __vsnprintf_internal (fs->p, avail, fmt, args, 0);
6 429 argp/argp-fmtstream.c <<<unknown>>>
    weak_alias (__argp_fmtstream_printf, argp_fmtstream_printf)
7 58 argp/argp-fmtstream.h <<<unknown>>>
    #define __argp_fmtstream_printf fprintf
8 59 argp/argp-fmtstream.h <<<unknown>>>
    #define argp_fmtstream_printf fprintf
-- More --
```

# Usage of cscope in vim

"cs find c printf":  
Prints the functions calling the symbol `printf`

```
Cscope tag: printf
#  line  filename / context / line
1    53  argp/argp-test.c <<weak_alias>>
      printf ("NO SUB ARGS\n");
2    53  argp/argp-test.c <<sub_parse_opt>>
      printf ("NO SUB ARGS\n");
3    56  argp/argp-test.c <<weak_alias>>
      printf ("SUB ARG: %s\n", arg);
4    56  argp/argp-test.c <<sub_parse_opt>>
      printf ("SUB ARG: %s\n", arg);
5    60  argp/argp-test.c <<weak_alias>>
      printf ("SUB KEY %c\n", key);
6    60  argp/argp-test.c <<sub_parse_opt>>
      printf ("SUB KEY %c\n", key);
7   132  argp/argp-test.c <<popt>>
      printf ("KEY %s: %s\n", buf, arg);
8   134  argp/argp-test.c <<popt>>
      printf ("KEY %s\n", buf);
-- More --
```



# Usage of cscope in vim

"cs find d printf":  
Print the functions called by the symbol printf.

```
Cscope tag: printf
# line filename / context / line
1 112 libio/bits/stdio2.h <<__printf_chk>>
    return __printf_chk (__USE_FORTIFY_LEVEL - 1, __fmt, __va_arg_pack ());
2 112 libio/bits/stdio2.h <<__va_arg_pack>>
    return __printf_chk (__USE_FORTIFY_LEVEL - 1, __fmt, __va_arg_pack ());
3 116 libio/bits/stdio2.h <<__printf_chk>>
    __printf_chk (__USE_FORTIFY_LEVEL - 1, __VA_ARGS__)
4 10 sysdeps/ieee754/ldbl-opt/nldbl-printf.c <<va_start>>
    va_start (arg, fmt);
5 11 sysdeps/ieee754/ldbl-opt/nldbl-printf.c <<__nldbl_vfprintf>>
    done = __nldbl_vfprintf (stdout, fmt, arg);
6 12 sysdeps/ieee754/ldbl-opt/nldbl-printf.c <<va_end>>
    va_end (arg);
7 104 /usr/include/bits/stdio2.h <<__printf_chk>>
    return __printf_chk (__USE_FORTIFY_LEVEL - 1, __fmt, __va_arg_pack ());
8 104 /usr/include/bits/stdio2.h <<__va_arg_pack>>
    return __printf_chk (__USE_FORTIFY_LEVEL - 1, __fmt, __va_arg_pack ());
-- More --
```



# Usage of cscope in vim

"cs find g printf":  
Search position defining the symbol printf

```
~
~
~
~
~
~
~
Cscope tag: printf
# line filename / context / line
1 110 libio/bits/stdio2.h <<printf>>
printf (const char *__restrict __fmt, ...)
2 115 libio/bits/stdio2.h <<printf>>
#define printf(...) \
3 5 sysdeps/ieee754/ldbl-opt/nldbl-printf.c <<printf>>
printf (const char *fmt, ...)
4 102 /usr/include/bits/stdio2.h <<printf>>
printf (const char *__restrict __fmt, ...)
5 107 /usr/include/bits/stdio2.h <<printf>>
#define printf(...) \
Type number and <Enter> (empty cancels): █
```

# make

Materials from EE485 taught by  
Youjip Won and Kyungsoo Park

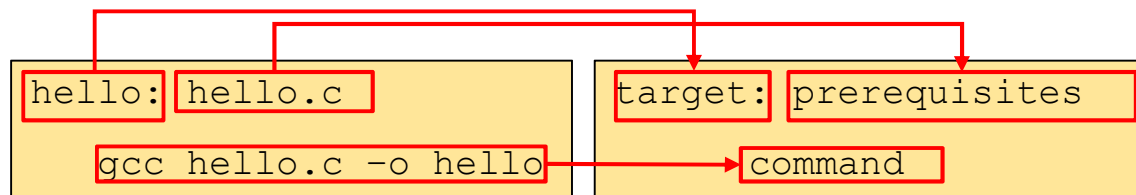
# What Is Makefile?

- A file that helps decide which parts of a large program need to be recompiled.
  - Makefile = a configuration file for partial build.
- Example: build a wordprocessor with 10,000 source code files.
  - Don't want to recompile them all if you edit just one line.
  - Recompile only those files that have changed.
  - make (a tool) enables this task.
    - A GNU tool that executes the “*appropriate*” rules in makefile.

```
$ make  
... (recompilation)
```

# make & makefile

- **make** executes rules in 'makefile' (or 'Makefile' or 'GNUMakefile').
- 'makefile' content:



"Run the target if the target file is **older** than the prerequisite files"  
=>target file is out of date

```
$ make hello // make takes a specific target for building
```

```
gcc hello.c -o hello
```

```
$ make // no target? Then, the first target (default target) is run.
```

# Targets and Prerequisites

- 'makefile' contains a set of rules to build an application.
  - Can have multiple rules in one file.
  - Each rule can have (multiple) dependencies.
- **Default** rule = the first rule in 'makefile'.
  - Default rule is executed if make runs without any arguments.
- A rule consists of a target, prerequisite(s), and command(s).

```
target: prereq1 prereq2
```

```
<tab> commands
```

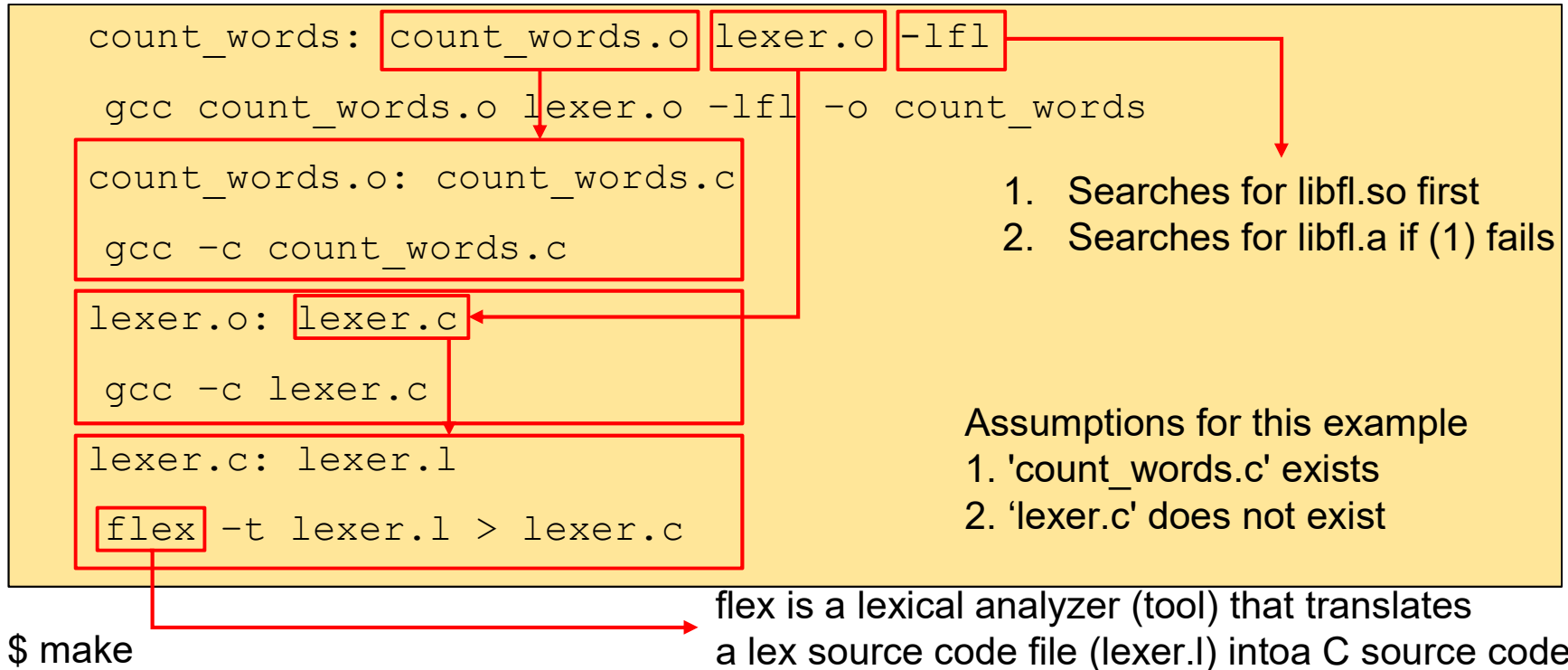
Important! no spaces other than a single tab ('\t') before the commands

# Example Rules in Makefile

```
foo: foo.o libfoo.o foo.h
    gcc foo.o libfoo.o -o foo
```

- *target*=foo, *prerequisites*=foo.o, libfoo.o, and foo.h
  - *command*=gcc foo.o libfoo.o -o foo
  - How to evaluate a rule?
    1. It finds the files for the prerequisites and the target.
    2. If the prerequisites have associated rules, evaluate them recursively.
- ```
foo.o: foo.c foo.h
    gcc -c foo.c
```
3. If any prerequisite is **newer** (modified more recently) than the target, the target is rebuilt by executing the command(s).
- Each command line is run in its own shell.
  - If any command fails, building of the target terminates and make exits.

# An Example for Evaluating Rules



```
gcc -c count_words.c
```

```
flex -t lexer.l > lexer.c
```

```
gcc -c lexer.c
```

```
gcc count_words.o lexer.o -lfl -o count_words
```

# A Few Tips on Rules

- A rule can have multiple targets.
- Each target has the same set of prerequisites.

```
vpath.o variable.o: make.h config.h getopt.h gettext.h dep.h
```



```
vpath.o: make.h config.h getopt.h gettext.h dep.h  
variable.o: make.h config.h getopt.h gettext.h dep.h
```

- Not all prerequisites need to be defined ***all at once***.

```
vpath.o: make.h config.h getopt.h gettext.h dep.h  
vapth.o: filedep.h hash.h job.h commands.h variable.h vapth.h
```

```
vpath.o: lexer.c
```

...

```
vpath.o: vpath.c
```

Whenever lexer.c is updated, vpath.o must be updated.  
The prerequisite is always updated before the target is updated.



# Wildcards and Variables

- `make` supports wildcards. (`*`, `~`, `?`, `[...]`, `[^...]`)
  - `*`: replaced with all file names in the current directory. (`*.c`: all `'c'` files in current directory)
  - `~`: replaced with the home directory.
- `$(variable-name)`: expand the variable whose name is `variable-name`.

- Format: `variable-name = value`
- Variables can contain almost any text

- Automatic variables

- `$@`: the filename that represents the target.
- `$<`: The filename of the first prerequisite.
- `$?`: the names of all prerequisites newer than the target, separated by spaces.
- `$^`: the filename of all the prerequisites, separated by spaces. (with duplicate files removed)
- `$+`: Similar to `$^`, but allows duplicate files.
- `$*`: the stem of the target filename, “stem” = a filename without its suffix.

```
CC = gcc
prog: *.c
    $(CC) -o $@ $^
```

 `gcc -o prog *.c`

# Practicing with Wildcards

```
count_words: count_words.o counter.o lexer.o -lfl
```

```
gcc $^ -o $@ → gcc count_words.o counter.o lexer.o -lfl -o count_words
```

```
count_words.o: count_words.c
```

```
gcc -c $< → gcc -c count_words.c
```

```
counter.o: counter.c
```

```
gcc -c $< → gcc -c counter.c
```

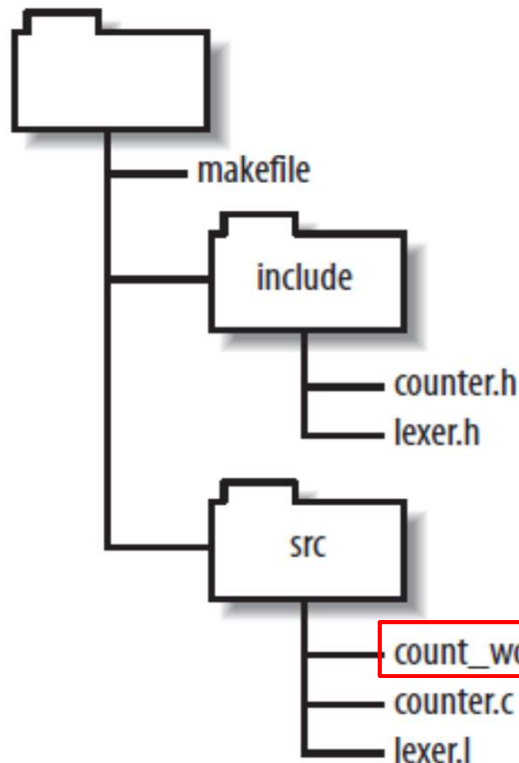
```
lexer.o: lexer.c
```

```
gcc -c $< → gcc -c lexer.c
```

```
lexer.c: lexer.l
```

```
flex -t $< > $@ → flex -t lexer.l > lexer.c
```

# Files in Different Directories?



```
count_words: count_words.o counter.o lexer.o -lfl
    gcc $^ -o $@
count_words.o: count_words.c include/counter.h
    gcc -c $<
counter.o: counter.c include/counter.h include/lexer.h
    gcc -c $<
lexer.o: lexer.c include/lexer.h
    gcc -c $<
lexer.c: lexer.l
    flex -t $< > $@
```

**\$ make**

make: \*\*\* No rule to make target 'count\_words.c',  
needed by 'count\_words.o'. Stop.

**How to fix this problem?**

Why? 'count\_words.c' is not found in the current directory (it's in src/) nor  
there is a rule to build the file. The error messages refers to the latter.

# VPATH & CPPFLAGS

```
VPATH = src

CPPFLAGS = -I include

count_words: count_words.o counter.o lexer.o -lfl
    gcc $(CPPFLAGS) $^ -o $@

count_words.o: count_words.c include/counter.h
    gcc $(CPPFLAGS) -c $<

counter.o: counter.c include/counter.h include/lexer.h
    gcc $(CPPFLAGS) -c $<

lexer.o: lexer.c include/lexer.h
    gcc $(CPPFLAGS) -c $<

lexer.c: lexer.l
    flex -t $< > $@
```

- VPATH: environment variable. tells make where to look for the files.
- CPPFLAGS: provides an option to gcc for finding the header files.
  - CPPFLAGS: preprocessor options
  - CFLAGS: c compiler options
- vpath directive – more precise

```
vpath %.c src
vpath %.l src
vpath %.h include
```

- Look for .c or .l files in “src/”.
- Look for .h in “include/” .
  - No need for include/X in prerequisites.

# Phony Targets

- Phony target: a target that does not represent a file.

- Always out of date, so always evaluate the rule.

```
clean:
    rm -f *.o lexer.c
```

- `$ make clean //` always executes `'rm -f *.o lexer.c'`.
- What if there happens to be a file, 'clean', in the current directory?

- `$ make clean`

make: 'clean' is up to date.

- Problem: make does not know whether a target is phony or not.
- `.PHONY` to explicitly tell make that 'clean' is a phony target.

```
.PHONY: clean
clean:
    rm -f *.o lexer.c
```

# Popular Phony Targets

---

- Typical phony targets & typical meaning
  - **all**: perform all tasks to build the application.
  - **install**: create an installation of the application from the compiled binaries.
  - **clean**: delete the binary files, temporary files generated from source files.
  - **distclean**: delete all the generated files not in the original source distribution.
  - **TAGS**: create a tag table for use by editors. (ctags/etags)
  - **info**: create GNU info files from their Texinfo sources.
  - **check**: Run any tests associated with this application.

# Pattern Rules

```
VPATH = src include
```

```
CPPFLAGS = -I include
```

```
count_words: counter.o lexer.o -lfl
```

```
count_words.o: counter.h
```

```
counter.o: counter.h lexer.h
```

```
lexer.o: lexer.h
```

```
%.o: %.c
```

```
$(COMPILE.c) $(OUTPUT_OPTION) $<
```

```
%.c: %.1
```

```
@$(RM) $@
```

```
$(LEX.1) $< > $@
```

```
%.o: %.o
```

```
$(LINK.o) $^ $(LOADLIBES) $(LDLIBS) -o $@
```

Assume variables below (COMPILE.c, RM, LEX.1, ...) are defined somewhere else

- Can simplify makefile with built-in rules for well-known file types.

- Built-in rules = **implicit** pattern rules

- Pattern rules = normal rules except the **stem** of the file is expressed as %.

a name without a suffix

- The first target expanded by the implicit pattern rule, '%: %.o'.

```
count_words: count_words.o counter.o lexer.o -lfl
```

- % = count\_words, %.o = count\_words.o

- Next target = count\_words.o, its rule expanded by '%.o: %.c'.

```
count_words.o: count_words.c counter.h
```

- src/count\_words.c exists, and it does not have a prerequisite => run the command & move on.

# Pattern Rules

```
VPATH = src include
CPPFLAGS = -I include
count_words: counter.o lexer.o -lfl
count_words.o: counter.h
counter.o: counter.h lexer.h
lexer.o: lexer.h
```

```
%.o: %.c
```

```
$(COMPILE.c) $(OUTPUT_OPTION) $<
```

```
%.c: %.l
```

```
@$(RM) $@
```

```
$(LEX.l) $< > $@
```

```
%.o: %.o
```

```
$(LINK.o) $^ $(LOADLIBES) $(LDLIBS) -o $@
```

- Next target = counter.o

```
counter.o: counter.c counter.h lexer.h
```

(expanded by **%.o: %.c**)

- Next target = lexer.o (expanded by **%.o: %.c**)

```
lexer.o: lexer.c lexer.h
```

- But src/lexer.c does **NOT** exist
- This triggers the rule, **%.c: %.**

```
lexer.c: lexer.l
```

```
@$(RM) $@
```

```
$(LEX.l) $< > $@
```



# Predefined Variables in Built-in Rules

```
% .o: %.c
    $(COMPILE.c) $(OUTPUT_OPTION) $<
%.c: %.l
    @$ (RM) $@
    $(LEX.l) $< > $@
%: %.o
    $(LINK.o) $^ $(LOADLIBES) $(LDLIBS) -o $@
```

- `$ make -p //` will show all predefined/default variables
  - `COMPILE.c` = `$(CC) $(CFLAGS) $(CPPFLAGS) $(TARGET_ARCH) -c`
  - `CC` = `cc`, can be redefined to the path of an alternate C compiler.
  - `CFLAGS`: options for `$(CC)` command. none by default.
  - `CPPFLAGS`: options for `cpp`. none by default .
  - `TARGET_ARCH`: architecture-specific options. none by default.
  - `LINK.o` = `$(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)`
  - `LDFLAGS`: options for `ld`; none by default.
  - `LEX.l` = `$(LEX) $(LFLAGS) -t`
  - `LEX` = `lex`
  - `LFLAGS` = options for `lex`, none by default.
  - `RM` = `rm -f`

# Practice Writing Makefile

- Want to build *progK*. (program binary name)
  - C source code consists of fileA.c, fileB.c, and fileC.c
  - fileA.c includes a1.h and a2.h, fileB.c includes b1.h, and fileC.c includes c1.h.
  - All files (.c or .h) can be modified any time.
- Let's write reasonable makfeile.
- Version 1:

```
all:
    gcc -c fileA.c fileB.c fileC.c -o progK
```

- Problem: no prerequisites – don't recompile even if any files are modified.
- Version 2:

```
all: fileA.c fileB.c fileC.c a1.h a2.h b1.h c1.h
    gcc -c fileA.c fileB.c fileC.c -o progK
```

- Problem? Recompile every file if any prerequisite files are modified.

# Practice Writing Makefile (Continued)

- Version 3:

```
OBJFILES = fileA.o fileB.o fileC.o
CC = gcc
CFLAGS = -Wall -Werror
All: $(OBJFILES)
    $(CC) $(OBJFILES) -o progK
fileA.o: a1.h a2.h
fileB.o: b1.h
fileC.o: c1.h
```

- A reasonable one that does
  - Partial/incremental build.
  - Properly uses built-in pattern rules.
  - Properly overrides CFLAGS with `-Wall` (print all warnings) `-Werror` (make all warnings into errors).

# Debugging is not easy.

In 1997, Pathfinder on Mars has stopped. OS has crashed due to the priority inversion.



- The Mars Pathfinder Mission Status Reports — First Week
- [The Mars Pathfinder Mission Status Reports — Second Week](#)
- The Mars Pathfinder Mission Status Reports — Third Week
- What really happened on Mars?
- [A Conversation with Glenn Reeves](#)

## How did NASA remotely fix the code on the Mars Pathfinder?



17



5

In 1997, NASA remotely fixed a bug that caused priority inversion on their Mars Pathfinder. How did they go about doing this? What kind of communication protocols are used? How do they update the source for an operating system, compile it, and run it from a remote location? This might be simpler than I thought, but to me this seems like quite the feat!

Story of the bugfix here: [http://research.microsoft.com/en-us/um/people/mbj/mars\\_pathfinder/authoritative\\_account.html](http://research.microsoft.com/en-us/um/people/mbj/mars_pathfinder/authoritative_account.html)

The author said to email him and he would provide details, but this was almost 20 years ago. Curious to see if anyone else knows how this worked.

asked :

viewed {

active :

Linked

2 E  
sj