System Programming

Prof. Chuan-Ju Wang
Dept. of Computer Science
University of Taipei

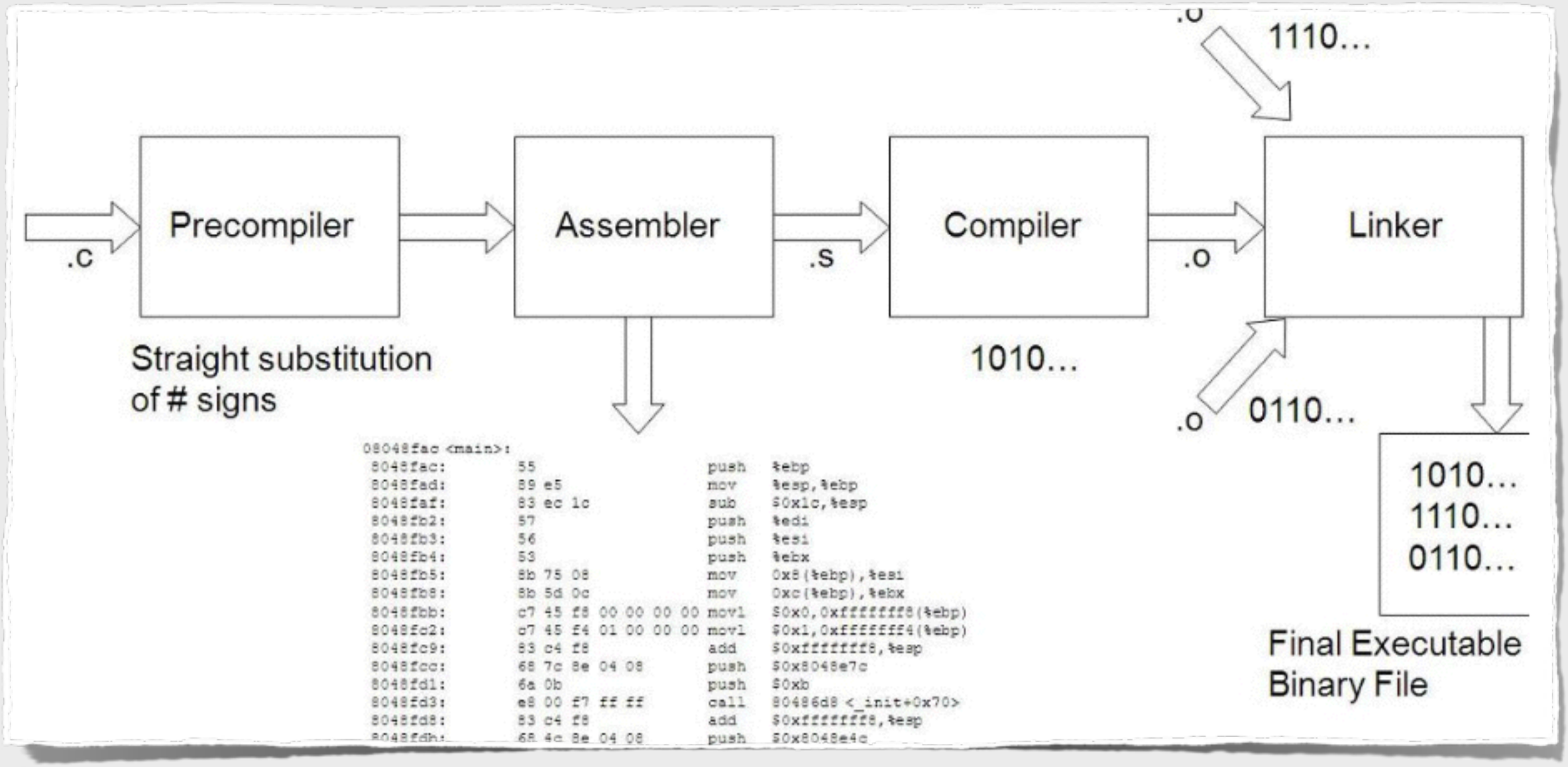
Compiler: gcc

- Command to compile a C program
  - `gcc -ansi -pedantic -Wall -Wextra -o main main.c`
- Options
  - **`-ansi`**: in C mode, support all ISO C90 programs
  - **`-pedantic`**: issue all the warnings demanded by strict ISO C
  - **`-Wall`**: enable all the warnings about constructions
  - **`-Wextra`**: print extra warning messages for several specific events
  - **`-o`**: outfile

Compiler: gcc

- Four stages to convert C code into an executable
  - **Pre-compiler**: processes **`.c`** files looking for **`#include`**, **`#define`** and other such macros and does text substitution etc.
  - **Assembler**: converts each **`.c`** file into an assembly language file with a **`.s`** extension
  - **Compiler**: coverts **`.s`** files into object (**`.o`**) files
  - **Linker**: combines the **`.o`** files and pulls in other compiled code from libraries (stdio, stdlib, string) that your program uses. It combines them to form a single binary executable file such as **`a.out`** or whatever you named it with the -o switch when you compiled.

Compiler: gcc




```
08048fac <main>:
 8048fac:       55                      push   %ebp
 8048fad:       89 e5                   mov    %esp,%ebp
 8048faf:       83 ec 1c                sub    $0x1c,%esp
 8048fb2:       57                      push   %edi
 8048fb3:       56                      push   %esi
 8048fb4:       53                      push   %ebx
 8048fb5:       8b 75 08                mov    0x8(%ebp),%esi
 8048fb8:       8b 5d 0c                mov    0xc(%ebp),%ebx
 8048fbb:       c7 45 f8 00 00 00 00    movl   $0x0,0xfffffff8(%ebp)
 8048fc2:       c7 45 f4 01 00 00 00    movl   $0x1,0xfffffff4(%ebp)
 8048fc9:       83 c4 f8                add    $0xfffffff8,%esp
 8048fcc:       68 7c 8e 04 08          push   $0x8048e7c
 8048fd1:       6a 0b                   push   $0xb
 8048fd3:       e8 00 f7 ff ff          call   80486d8 <_init+0x70>
 8048fd8:       83 c4 f8                add    $0xfffffff8,%esp
 8048fdb:       68 4c 8e 04 08          push   $0x8048e4c
```

Compiler: gcc

- Intermediate stage
  - **Assembler**: Compiling a **`.c`** file into a **`.s`** (assembly language) file by using the **`-S`** switch
  - **Object Files**
    - Produced by compilation but are typically deleted by the linker
    - The automatic destruction of the object file can be suppressed by suppressing the linker with the **`-c`** switch.

Compiler: gcc

- HelloWorld.c
  - `gcc -S -ansi -pedantic -W -Wall HelloWorld.c`

HelloWorld.s (~/test) - VIM

HelloWorld.s

```
1     .file   "HelloWorld.c"
2     .section    .rodata
3 .LC0:
4     .string "Hello World"
5     .text
6     .globl  main
7     .type   main, @function
8 main:
9 .LFB0:
10    .cfi_startproc
11    pushl   %ebp
12    .cfi_def_cfa_offset 8
13    .cfi_offset 5, -8
14    movl    %esp, %ebp
15    .cfi_def_cfa_register 5
16    andl    $-16, %esp
17    subl    $16, %esp
18    movl    $.LC0, %eax
```

Compiler: gcc

- HelloWorld.c
  - `gcc -c -ansi -pedantic -W -Wall HelloWorld.c`

```
jere@VirtualBox-MBP [~/test] gcc -c -ansi -pedantic -W -Wall HelloWorld.c
jere@VirtualBox-MBP [~/test] ls *.o
HelloWorld.o
jere@VirtualBox-MBP [~/test]
```

Execute the Program

```
jere@VirtualBox-MBP [~/test] gcc -ansi -pedantic -W -Wall -o HW HelloWorld.c
jere@VirtualBox-MBP [~/test] ls -al
total 48
drwxrwxr-x  2 jere jere 4096 2012-01-02 16:39 .
drwxr-xr-x 23 jere jere 4096 2012-01-02 16:38 ..
-rwxrwxr-x  1 jere jere 7166 2012-01-02 16:38 HelloWorld
-rw-rw-r--  1 jere jere  105 2012-01-02 16:38 HelloWorld.c
-rw-rw-r--  1 jere jere  103 2012-01-02 16:31 HelloWorld.c~
-rw-rw-r--  1 jere jere 1040 2012-01-02 16:31 HelloWorld.o
-rw-rw-r--  1 jere jere  514 2012-01-02 16:37 HelloWorld.s
-rw-rw-r--  1 jere jere  514 2012-01-02 16:35 HelloWorld.s~
-rwxrwxr-x  1 jere jere 7164 2012-01-02 16:39 HW
-rw-rw-r--  1 jere jere   31 2012-01-02 16:00 test.cpp
jere@VirtualBox-MBP [~/test] ./HW
Hello World
```

Makefile

- Makefile
  - Specify how to derive the target program
  - Becomes a necessity when working with larger programs consisting of many source files and thousands of lines of code

Makefile

- Example1

```
# 1. 先進行目標檔的編譯，最終會有四個 *.o 的檔名出現：
[root@www ~]# gcc -c main.c
[root@www ~]# gcc -c haha.c
[root@www ~]# gcc -c sin_value.c
[root@www ~]# gcc -c cos_value.c

# 2. 再進行連結成為執行檔，並加入 libm 的數學函式，以產生 main 執行檔：
[root@www ~]# gcc -o main main.o haha.o sin_value.o cos_value.o \
> -lm -L/usr/lib -L/lib

# 3. 本程式的執行結果，必須輸入姓名、360 度角的角度值來計算：
[root@www ~]# ./main
Please input your name: VBird   <==這裡先輸入名字
Please enter the degree angle (ex> 90): 30    <==輸入以 360 度角為主的角度
Hi, Dear VBird, nice to meet you.     <==這三行為輸出的結果喔！
The Sin is:  0.50
The Cos is:  0.87
```

* Library usage: http://www.vr.ncue.edu.tw/esa/EmbeddedSystemProgramming2010/ch09.htm

Makefile

- Using Makefile

+ Makefile

```
1 main: main.o haha.o sin_value.o cos_value.o
2     gcc -o main main.o haha.o sin_value.o cos_value.o -lm
3
4 clean:
5     rm *.o main
```

```
[root@www ~]# make
cc     -c -o main.o main.c
cc     -c -o haha.o haha.c
cc     -c -o sin_value.o sin_value.c
cc     -c -o cos_value.o cos_value.c
gcc -o main main.o haha.o sin_value.o cos_value.o -lm
# 此時 make 會去讀取 makefile 的內容，並根據內容直接去給他編譯相關的檔案囉！

# 3. 在不刪除任何檔案的情況下，重新執行一次編譯的動作：
[root@www ~]# make
make: `main' is up to date.
# 看到了吧！是否很方便呢！只會進行更新 (update) 的動作而已。
```

Makefile

- Variable Definitions

```
CFLAGS = -ansi -pedantic -Wall
SRCS = main.c file1.c file2.c
CC = gcc
```

- Dependency Rules

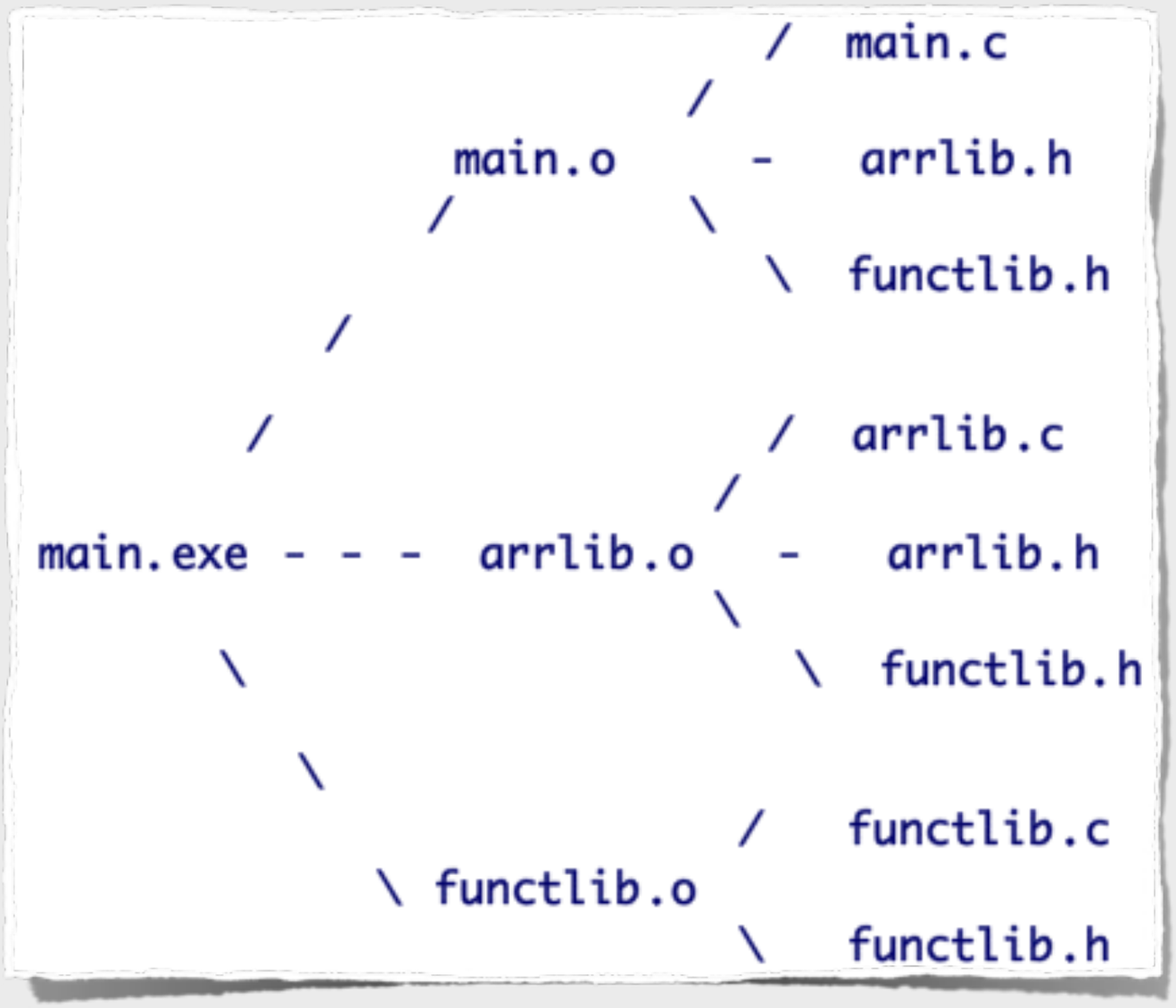
```
target:[dependencies]
<Tab><command>
<Tab><command 2>
    ...
```

```
    main.o: main.c
        $(CC) $(FLAGS) -c main.c
    clean:
        rm *.o
```

- Comments: Any line beginning with a "#" sign, or any line that contains only white-space.

Makefile

- Example2



Makefile

```
1  # Make file that builds executable file main.exe
2
3  CC= gcc
4  CFLAGS= -ansi -pedantic -Wall
5
6  main.exe: main.o arrlib.o functlib.o
7      $(CC) $(CFLAGS) -o main.exe main.o arrlib.o functlib.o
8
9  main.o: main.c arrlib.h functlib.h
10     $(CC) $(CFLAGS) -c main.c
11
12 arrlib.o: arrlib.c arrlib.h functlib.h
13     $(CC) $(CFLAGS) -c arrlib.c
14
15 functlib.o: functlib.c  functlib.h
16     $(CC) $(CFLAGS) -c functlib.c
17 clean:
18     rm main.exe *.o
```

Makefile

- Resources
  - Makefiles
    - http://www.cprogramming.com/tutorial/makefiles.html
    - http://linux.vbird.org/linux_basic/0520source_code_and_tarball.php#make_why
    - http://maxubuntu.blogspot.com/2010/02/makefile.html
  - Automating Program Compilation - Writing Makefiles
    - http://users.actcom.co.il/~choo/lupg/tutorials/writing-makefiles/writing-makefiles.html

Bash (Unix Shell)

- There are two main types of shells: Graphical User Interface (GUI), and Command Line Interface (CLI).
- Bash is used by many Linux distributions as the default CLI shell.
- Bash can be used not only as a user interface to the operating system, but also as a programming environment.
- Bash is an acronym for Bourne Again SHell, named after Steve Bourne's shell (released for UNIX in 1979).

```
jere@VirtualBox-MBP [~/test] cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
colord:x:102:105:colord colour management daemon,,,:/var/lib/colord:/bin/false
messagebus:x:103:107::/var/run/dbus:/bin/false
lightdm:x:104:108:Light Display Manager:/var/lib/lightdm:/bin/false
avahi-autoipd:x:105:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:106:113:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
kernoops:x:108:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:109:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:110:122:RealtimeKit,,,:/proc:/bin/false
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
hplip:x:112:7:HPLIP system user,,,:/var/run/hplip:/bin/false
saned:x:113:123::/home/saned:/bin/false
jere:x:1000:1000:jere,,,:/home/jere:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
```

Why Bash?

- While there are other shells available, Bash has a number of distinct advantages:
  - Command Line Editing (Go back and fix typos, utilize history)
  - Tab Completion Job Control (Start, stop, pause, and background jobs)
  - Customization (For advanced users)
  - Bash is Free and Open Source Software, distributed under the GPL

Bash Basics

Type **exit/logout** **ctrld**	exit
Type **clear** ctrll (L)	clear the screen
Ctrlc	stop current command
Ctrl\	stop current command (more forceful than ctrlc)
Ctrls	pause output to the screen
Ctrlq	restart output to the screen
Ctrlu	erase current command line
Tab	auto complete current command or filename

Simple Clean-Up Example

- Assume that I need to clean up a directory every time before executing a program

```
$ rm –rf *.stderr
$ rm –rf *.stdout
$ rm –rf *.tmp
```

Shell scripts

- List of command, executed in order
  - `#!`: tells the CPU what shell to use to execute script
  - The shell name is the shell that will execute this script.
    - E.g., `#!/bin/bash` (which we will use)
- If no shell is specified in the script file, the default is chosen to be the executing shell.

The First Bash Script

- Write programs using **`vi`**
- So fire up a text editor; for example:

```
$ mkdir ~/scripts
$ cd scripts
$ vi hello.sh
```

- Type the following inside it:

```
#!/bin/bash
# This is a commented line, will not be executed
# This is my first script "Hello World"
echo "Hello World"
```

- Make the script executable:

```
$chmod u+x hello.sh
$ls –l
-rwxr--r-- hello.sh
```

The First Bash Script

- To execute the program:

```
$ hello.sh
-bash: hello.sh: command not found
```

  - $PATH environment variable holds the location where all commands are stored

```
$ echo $PATH
/usr/bin:/bin:/usr/sbin
```

- We must specify the path of hello.sh

```
$ /cchome/arodrigu1/scripts/hello.sh
$ ./hello.sh
Hello World
```

Back to the Clean-Up Example

- We can put all those commands into a shell script, called mycleanDir.sh.

```
$ vi mycleanDir.sh
#!/bin/bash
rm –rf *.stderr
rm –rf *.stdout
rm –rf *.tmp
echo "Deleted files with suffix blastout, stderr, stdout, tmp"
```

Make it executable and run!

Variables

- There are two types of variables
  - Environmental variables
  - Local variables

Environmental Variables

- Environmental variables hold special values.
- Environmental variables are set by the system on initial login
  - **`/etc/profile`**, **`/etc/profile.d/`** and **`~/.bash_profile`** or **`~/.profile`**.

- If you want to know what the variable holds call it with a "**$**" sign:
- `env` command

```
$ echo SHELL
SHELL
$ echo $SHELL
/bin/bash
$ echo $HOME
/cchome/arodrigu1
$ echo $PATH
/usr/X11R6/bin:/usr/local/bin:/bin:/
usr/bin
```

Ubuntu [Running]

jere@jere-VirtualBox: ~/test/script

```
SSH_AGENT_PID=1457
GPG_AGENT_INFO=/tmp/keyring-pPNXAg/gpg:0:1
TERM=xterm
SHELL=/bin/bash
XDG_SESSION_COOKIE=b9e0472ea390e1346b6eb81a00000008-1323772057.86708-1990901206
WINDOWID=58720261
GNOME_KEYRING_CONTROL=/tmp/keyring-pPNXAg
GTK_MODULES=canberra-gtk-module:canberra-gtk-module
USER=jere
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:su=
34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01
1:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*
jar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35
:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=0
01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.
nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35
.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;3
6:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.axa=00;36:*.oga=0
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/tmp/keyring-pPNXAg/ssh
SESSION_MANAGER=local/jere-VirtualBox:@/tmp/.ICE-unix/1395,unix/jere-VirtualBox:/tmp/.ICE-unix/1395
USERNAME=jere
DEFAULTS_PATH=/usr/share/gconf/ubuntu-2d.default.path
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu-2d:/etc/xdg
PATH=/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
DESKTOP_SESSION=ubuntu-2d
PWD=/home/jere/test/script
GNOME_KEYRING_PID=1386
LANG=en_US.UTF-8
MANDATORY_PATH=/usr/share/gconf/ubuntu-2d.mandatory.path
UBUNTU_MENUPROXY=libappmenu.so
```

Environmental Variables

- **`PATH`**: The search path for commands
- Usually, we type in the commands in the following way:

```
$ ./hello.sh
Hello World
```

- By setting `PATH=$PATH:~/scripts` our working directory is included in the search path for commands, and we simply use the export command:

```
$ export PATH=$PATH:~/scripts
$ hello.sh
Hello World
```

Local Variables

- We can use variables as in any programming languages
- Stored as strings
- Declaring a variable:

```
$ STR='Hello World!'
$ echo $STR
Hello World!
```

- a value to a variable
- Call the variable by putting the '**$**' at the beginning

Double Quotes

- When assigning character data containing spaces or special characters, the data must be enclosed in either single or double quotes.
- Using double quotes (partial quoting) to show a string of characters will allow any variables in the quotes to be resolved.

```
$ var="test string"
$ newvar="Value of var is $var"
$ echo $newvar
Value of var is test string
```

Single Quotes

- Using single quotes (full quoting) to show a string of characters will not allow variable resolution.

```
$ newvar='Value of var is $var'
$ echo $newvar
Value of var is $var
```

Command Substitution

- The backquote “`” is different from the single quote “’“.
  - It is used for command substitution: `command`
- You can assign the output of a command to a variable.

```
$ ls
hello.sh myCleanDir.sh
$ LIST=`ls`
$ echo $LIST
hello.sh myCleanDir.sh
```

Conditional Statements

- Conditionals lets us decide whether to perform an action or not, this decision is taken by evaluating an expression

```
if [ expression ];       ## must have space between brackets
then
  statements
elif [ expression ];     ## brackets test an expression
then
  statememts
else
  statements
fi
```

- The **`elif`** (else if) and **`else`** sections are optional.

Conditional Statements - Example

- Lets write a script that determines whether the word “UNIX” exists in the file "myfile"

grep: returns 0 if it finds something; returns non-zero otherwise

```
$ vi if1.sh
if grep "UNIX" myfile >/dev/null
then
  echo "It's there"
fi
$ ./if1.sh
It's there
```

redirect to **/dev/null** so that "intermediate" results do not get printed
This file is available for everyone

Conditional Statements - Example

```
$ vi if2.sh
#!/bin/bash
if grep "UNIX" myfile2 >/dev/null
then
  echo "UNIX occurs in myfile2"
else
  echo "No!"
  echo "UNIX does not occur in myfile2"
fi
$ ./if2.sh
No! UNIX does not occur in myfile2
```

Expressions

- Expressions can be:
  - String comparison
  - Numeric comparison
  - File operators
  - Logical operators

Expressions: String Comparisons

- String Comparisons:
  - `=` compare if two strings are equal
  - `!=` compare if two strings are not equal
  - `-n` evaluate if string length is greater than zero
  - `-z` evaluate if string length is equal to zero

Expressions: String Comparisons

- Examples:
  - `[ s1 = s2 ]` (true if s1 same as s2, else false)
  - `[ s1 != s2 ]` (true if s1 not same as s2, else false)
  - `[ s1 ]` (true if s1 is not empty, else false)
  - `[ -n s1 ]` (true if s1 has a length greater then 0, else false)
  - `[ -z s2 ]` (true if s2 has a length of 0, otherwise false)

Expressions: String Comparisons

- Compare the user's name given with the environment variable **$USER**

```
$ vi if3.sh
#!/bin/bash
echo -n "Enter your login name: "  # ask user input
read name # store input in var
if [ "$name" = "$USER" ];
then
  echo "Hello, $name. How are you today ?"
else
  echo "You are not $USER, so who are you ?"
fi
$ ./if3.sh
Enter your login name: Jackie
You are not jere, so who are you ?
```

Expressions: Number Comparisons

- Number Comparisons:
  - **`-eq`** compare if two numbers are equal
  - **`-ge`** compare if one number is greater than or equal to a number
  - **`-le`** compare if one number is less than or equal to a number
  - **`-ne`** compare if two numbers are not equal
  - **`-gt`** compare if one number is greater than another number
  - **`-lt`** compare if one number is less than another number

Expressions: Number Comparisons

- Examples:
  - `[ n1 -eq n2 ]` (true if n1 same as n2, else false)
  - `[ n1 -ge n2 ]` (true if n1 greater than or equal to n2, else false)
  - `[ n1 -le n2 ]` (true if n1 less then or equal to n2, else false)
  - `[ n1 -ne n2 ]` (true if n1 is not same as n2, else false)
  - `[ n1 -gt n2 ]` (true if n1 greater then n2, else false)
  - `[ n1 -lt n2 ]` (true if n1 less then n2, else false)

Expressions: Number Comparisons

- Perform a mathematical operation if the number is between a range, otherwise let the user know the number entered is incorrect

```
$ vi if4.sh
#!/bin/bash
echo-n"Enteranumber1<x<10:" #ask user input
read num  # store input in var
if [ "$num" -lt 10 ]; then
  if [ "$num" -gt 1 ]; then
    echo "$num*$num=$(($num*$num))"
  else
    echo "Wrong insertion !"
  fi
else
  echo "Wrong insertion !"
fi
$ ./if4.sh
Enter a number 1 < x < 10: 5
5*5=25
```

Expressions: File Operators

- Files operators:
  - `-d` check if path given is a directory
  - `-f` check if path given is a file
  - `-s` check if path given is a symbolic link
  - `-e` check if file name exists
  - `-s` check if a file has a length greaterthan0
  - `-r` check if read permission is set for file or directory
  - `-w` check if write permission is set for a file or directory
  - `-x` check if execute permission is set for a file or directory

Expressions: File Operators

- Check if a certain file exists

```
$ vi if5.sh
#!/bin/bash
if [ -f /etc/fstab ]; then
  cp /etc/fstab .
  echo "Done."
else
  echo "This file does not exist."
  exit 1
fi
$ ./if5.sh
Done.
```

Expressions: Logical Operators

- Logical operators:
  - **&&** logically AND two logical expressions
  - **||** logically OR two logical expressions

Expressions: Logical Operators

- The numerical example (if6.sh) can be made into one if statement by using logical operators.

```
$ vi if6.sh
#!/bin/bash
echo -n "Enter a number 1 < x < 10: "
read num
if [ "$num" -gt 1 ] && [ "$num" -lt 10 ]; then
  echo "$num*$num=$(($num*$num))"
else
  echo "Wrong insertion !"
fi
$ ./if6.sh
Enter a number 1 < x < 10: 5
5*5=25
```

for Loops

- Syntax:

```
for var in value1 value2 ...
do
  command_set
done
```

for Loops

- Lets calculate the smallest number among a set

```
$ vi for1.sh
#!/bin/bash
smallest=10000
for i in 5 8 19 8 7 3
do
  if [ $i -lt $smallest ]
  then
    smallest=$i
  fi done
echo $smallest
$ ./for1.sh
3
```

while Loops

- Syntax:

```
while [ expression ]
do
  command_set
done
```

while loop

- Lets do a summation of every number from 1 to 100

```
$ vi while1.sh
#! /bin/bash
i=1 # declare var
sum=0 # declare var
while [ $i -le 100 ]
do
  sum=`expr $sum + $i`
  i=`expr $i + 1`
done
echo The sum is $sum.
$ ./while1.sh
The sum is 5050.
```

Login Shell and Non-Login Shell

- When you login your system and see the command line prompt, it's a login shell, and it executes these files in order:
  - /etc/profile
  - ~/.bash_profile
  - ~/.bash_login
  - ~/.profile
- A non-login shell will only execute the two files in order:
  - /etc/bashrc
  - ~/.bashrc

* Login and non-login shell: http://linux.vbird.org/linux_basic/0320bash.php#settings_bashrc

Some Examples

- for_example.sh
- mkdirs.sh
- rename.sh
- count.sh
- average.sh

Task 3: Customize Your Prompt

```
jere@VirtualBox-MBP [~]
jere@VirtualBox-MBP [~] ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public
jere@VirtualBox-MBP [~]
jere@VirtualBox-MBP [~]
```

Customize your Prompt to
Your_ID@VirtualBox
Try to make it colorful!

Ex: u9916001@VirtualBox

- Hint: Revise ~/.bashrc

Task 4:
Compiling Source Files in the Book

```
jere@VirtualBox-MBP [~] mkdir SystemProgramming
jere@VirtualBox-MBP [~] ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  shared  SystemProgramming
jere@VirtualBox-MBP [~] cd SystemProgramming/
jere@VirtualBox-MBP [~/SystemProgramming] ls
jere@VirtualBox-MBP [~/SystemProgramming] wget http://www.apuebook.com/src.tar.gz
--2012-01-09 15:55:49--  http://www.apuebook.com/src.tar.gz
Resolving www.apuebook.com... 64.71.34.90
Connecting to www.apuebook.com|64.71.34.90|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 106578 (104K) [application/x-gzip]
Saving to: `src.tar.gz'

100%[======================================================================================>]

2012-01-09 15:55:50 (123 KB/s) - `src.tar.gz' saved [106578/106578]

jere@VirtualBox-MBP [~/SystemProgramming] ls
src.tar.gz
jere@VirtualBox-MBP [~/SystemProgramming] tar zxf src.tar.gz
jere@VirtualBox-MBP [~/SystemProgramming] ls
apue.2e  src.tar.gz
jere@VirtualBox-MBP [~/SystemProgramming]
```

http://www.apuebook.com/src.3e.tar.gz

* Change to: src.2e.tar.gz

Task 4:
Compiling Source Files in the Book

```
jere@VirtualBox-MBP [~/SystemProgramming] cd apue.2e/
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e] ls
advio        fig10.25  fig12.12  fig14.5   fig16.16  fig17.25  fig18.12  fig2.15  fig5.5   fig8.28  figC.21                  opend.fe
call         fig10.26  fig12.13  fig14.6   fig16.17  fig17.26  fig18.13  fig2.16  fig6.2   fig8.29  figC.3                   open.fe
calld        fig10.28  fig12.16  fig14.7   fig16.18  fig17.27  fig18.14  fig3.1   fig7.1   fig8.3   figC.4                   proc
daemons      fig10.29  fig12.17  fig14.9   fig16.20  fig17.28  fig18.15  fig3.10  fig7.11  fig8.30  figC.5                   pty
datafiles    fig10.30  fig12.4   fig1.5    fig16.8   fig17.29  fig18.16  fig3.11  fig7.13  fig8.5   figC.6                   README
db           fig10.5   fig12.8   fig15.11  fig16.9   fig17.30  fig18.17  fig3.2   fig7.14  fig8.6   figC.7                   sess
DISCLAIMER   fig10.6   fig1.3    fig15.12  fig1.7    fig17.31  fig18.18  fig3.4   fig7.16  fig8.8   figC.9                   signals
environ      fig10.7   fig13.1   fig15.14  fig17.10  fig17.32  fig18.20  fig4.12  fig7.3   fig9.11  file                     sockets
exercises    fig10.8   fig13.6   fig15.15  fig17.11  fig17.33  fig18.21  fig4.16  fig7.4   figB.1   include                  std
fig10.10     fig10.9   fig13.7   fig15.17  fig17.12  fig17.34  fig18.22  fig4.21  fig7.9   figB.3   ipc                      stdio
fig10.11     fig1.10   fig13.8   fig15.18  fig17.13  fig17.35  fig1.9    fig4.22  fig8.1   figB.4   ipp                      streams
fig10.12     fig11.10  fig1.4    fig15.19  fig17.14  fig17.36  fig19.10  fig4.23  fig8.12  figC.1   lib                      systype.sh
fig10.14     fig11.11  fig14.1   fig15.31  fig17.15  fig17.37  fig19.11  fig4.24  fig8.13  figC.11  lock                     termios
fig10.15     fig11.12  fig14.12  fig15.33  fig17.16  fig17.38  fig19.12  fig4.25  fig8.16  figC.12  Make.defines.freebsd     threadctl
fig10.18     fig11.13  fig14.16  fig15.5   fig17.17  fig17.39  fig19.13  fig4.3   fig8.17  figC.13  Make.defines.linux       threads
fig10.19     fig11.14  fig14.17  fig15.6   fig17.19  fig17.4   fig19.17  fig4.8   fig8.20  figC.14  Make.defines.macos
fig10.2      fig11.2   fig14.18  fig15.7   fig17.20  fig17.40  fig19.8   fig4.9   fig8.21  figC.15  Make.defines.solaris
fig10.20     fig11.3   fig14.19  fig1.6    fig17.21  fig17.6   fig19.9   fig5.11  fig8.22  figC.16  Makefile
fig10.22     fig11.4   fig14.29a fig16.10  fig17.22  fig1.8    fig20.3   fig5.12  fig8.23  figC.17  mycat
fig10.23     fig11.5   fig14.29b fig16.14  fig17.23  fig18.10  fig2.12   fig5.13  fig8.24  figC.18  open
fig10.24     fig12.11  fig14.32  fig16.15  fig17.24  fig18.11  fig2.13   fig5.4   fig8.25  figC.20  opend
```

What is sudo?
https://help.ubuntu.com/community/RootSudo

```
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e]
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e] cd include/
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e/include] ls
apue.h
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e/include] cp apue.h /usr/include/
cp: cannot create regular file `/usr/include/apue.h': Permission denied
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e/include] sudo cp apue.h /usr/include/
[sudo] password for jere:
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e/include] cd /usr/include/
jere@VirtualBox-MBP [/usr/include] ls
aio.h          ctype.h      fmtmsg.h         ifaddrs.h    mm               netpacket    python2.7
aliases.h      dbus-1.0     fnmatch.h        inttypes.h   mntent.h         netrom       rdma
alloca.h       dirent.h     fstab.h          langinfo.h   monetary.h       netrose      re_comp.h
a.out.h        dlfcn.h      fts.h            lastlog.h    mqueue.h         nfs          regex.h
apue.h         drm          ftw.h            libgen.h     mtd              nl_types.h   regexp.h
argp.h         elf.h        _G_config.h      libintl.h    nautilus-sendto  nss.h        resolv.h
argz.h         endian.h     gconv.h          libio.h      net              obstack.h    rpc
ar.h           envz.h       getopt.h         limits.h     netash           paths.h      rpcsvc
arpa           err.h        glob.h           link.h       netatalk         poll.h       sched.h
asm-generic    errno.h      gnu-versions.h   linux        netax25          printf.h     scsi
assert.h       error.h      grp.h            locale.h     netdb.h          protocols    search.h
byteswap.h     execinfo.h   gshadow.h        malloc.h     neteconet        pthread.h    semaphore.h
complex.h      fcntl.h      i386-linux-gnu   math.h       netinet          pty.h        setjmp.h
cpio.h         features.h   iconv.h          mcheck.h     netipx           pwd.h        sgtty.h
crypt.h        fenv.h       ieee754.h        memory.h     netiucv          python2.6    shadow.h
jere@VirtualBox-MBP [/usr/include] cd ~
jere@VirtualBox-MBP [~] cd SystemProgramming/
jere@VirtualBox-MBP [~/SystemProgramming] cd apue.2e/
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e]
```

Task 4:
Compiling Source Files in the Book

```
jere@VirtualBox-MBP [~/SystemProgramming] cd apue.2e/
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e] ls
advio       fig10.25  fig12.12  fig14.5   fig16.16  fig17.25  fig18.12  fig2.15  fig5.5   fig8.28  figC.21
call        fig10.26  fig12.13  fig14.6   fig16.17  fig17.26  fig18.13  fig2.16  fig6.2   fig8.29  figC.3
calld       fig10.28  fig12.16  fig14.7   fig16.18  fig17.27  fig18.14  fig3.1   fig7.1   fig8.3   figC.4
daemons     fig10.29  fig12.17  fig14.9   fig16.20  fig17.28  fig18.15  fig3.10  fig7.11  fig8.30  figC.5
datafiles   fig10.30  fig12.4   fig1.5    fig16.8   fig17.29  fig18.16  fig3.11  fig7.13  fig8.5   figC.6
db          fig10.5   fig12.8   fig15.11  fig16.9   fig17.30  fig18.17  fig3.2   fig7.14  fig8.6   figC.7
DISCLAIMER  fig10.6   fig1.3    fig15.12  fig1.7    fig17.31  fig18.18  fig3.4   fig7.16  fig8.8   figC.9
environ     fig10.7   fig13.1   fig15.14  fig17.10  fig17.32  fig18.20  fig4.12  fig7.3   fig9.11  file
exercises   fig10.8   fig13.6   fig15.15  fig17.11  fig17.33  fig18.21  fig4.16  fig7.4   figB.1   include
fig10.10    fig10.9   fig13.7   fig15.17  fig17.12  fig17.34  fig18.22  fig4.21  fig7.9   figB.3   ipc
fig10.11    fig1.10   fig13.8   fig15.18  fig17.13  fig17.35  fig1.9    fig4.22  fig8.1   figB.4   ipp
fig10.12    fig11.10  fig1.4    fig15.19  fig17.14  fig17.36  fig19.10  fig4.23  fig8.12  figC.1   lib
fig10.14    fig11.11  fig14.1   fig15.31  fig17.15  fig17.37  fig19.11  fig4.24  fig8.13  figC.11  lock
fig10.15    fig11.12  fig14.12  fig15.33  fig17.16  fig17.38  fig19.12  fig4.25  fig8.16  figC.12  Make.defin
fig10.18    fig11.13  fig14.16  fig15.5   fig17.17  fig17.39  fig19.13  fig4.3   fig8.17  figC.13  Make.defin
fig10.19    fig11.14  fig14.17  fig15.6   fig17.19  fig17.4   fig19.17  fig4.8   fig8.20  figC.14  Make.defin
fig10.2     fig11.2   fig14.18  fig15.7   fig17.20  fig17.40  fig19.8   fig4.9   fig8.21  figC.15  Make.defin
fig10.20    fig11.3   fig14.19  fig1.6    fig17.21  fig17.6   fig19.9   fig5.11  fig8.22  figC.16  Makefile
fig10.22    fig11.4   fig14.29a fig16.10  fig17.22  fig1.8    fig20.3   fig5.12  fig8.23  figC.17  mycat
fig10.23    fig11.5   fig14.29b fig16.14  fig17.23  fig18.10  fig2.12   fig5.13  fig8.24  figC.18  open
fig10.24    fig12.11  fig14.32  fig16.15  fig17.24  fig18.11  fig2.13   fig5.4   fig8.25  figC.20  opend
jere@VirtualBox-MBP [~/SystemProgramming/apue.2e] make
```

You may get an error like:
/usr/bin/ld: cannot find -lbsd
How to resolve it?

Task 4:
Compiling Source Files in the Book

```
jere@MacBook:~/SystemProgramming/apue.3e/intro
^_^jere@MBPR [~/.../apue.3e]
^_^jere@MBPR [~/.../apue.3e]
^_^jere@MBPR [~/.../apue.3e] ls -al
total 148
drwxr-xr-x 28 jere jere 4096 2014-02-23 21:54 ./
drwxrwxr-x  3 jere jere 4096 2014-02-23 21:44 ../
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 advio/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 daemons/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 datafiles/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 db/
-r--r--r--  1 jere jere  656 2013-04-27 10:18 DISCLAIMER
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 environ/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 exercises/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:44 figlinks/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 filedir/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 fileio/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:44 include/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:55 intro/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 ipc1/
drwxr-xr-x  6 jere jere 4096 2014-02-23 21:48 ipc2/
drwxr-xr-x  2 jere jere 4096 2014-02-23 21:48 lib/
```

```
jere@MacBook:~/SystemProgramming/apue.3e/intro
fig10.22  fig11.10  fig12.17  fig14.24  fig15.31  fig16.20  fig17.20  fig17.5
fig10.23  fig11.11  fig12.4   fig14.27  fig15.33  fig16.22  fig17.21  fig17.8
fig10.24  fig11.12  fig12.8   fig14.5   fig15.35  fig16.9   fig17.22  fig17.9
fig10.25  fig11.14  fig1.3    fig14.6   fig15.5   fig1.7    fig17.23  fig1.8
fig10.26  fig11.15  fig13.1   fig14.7   fig15.6   fig17.10  fig17.24  fig18.10
^_^jere@MBPR [~/SystemProgramming/.../figlinks] ls -al
total 8
drwxr-xr-x  2 jere jere 4096 2014-02-23 22:18 ./
drwxr-xr-x 28 jere jere 4096 2014-02-23 21:54 ../
lrwxrwxrwx  1 jere jere   18 2013-04-27 12:31 fig10.10 -> ../signals/read1.c
lrwxrwxrwx  1 jere jere   18 2013-04-27 12:31 fig10.11 -> ../signals/read2.c
lrwxrwxrwx  1 jere jere   19 2013-04-27 12:31 fig10.12 -> ../signals/setops.c
lrwxrwxrwx  1 jere jere   15 2013-04-27 12:31 fig10.14 -> ../lib/prmask.c
lrwxrwxrwx  1 jere jere   21 2013-04-27 12:31 fig10.15 -> ../signals/critical.c
lrwxrwxrwx  1 jere jere   15 2013-04-27 12:31 fig10.18 -> ../lib/signal.c
lrwxrwxrwx  1 jere jere   19 2013-04-27 12:31 fig10.19 -> ../lib/signalintr.c
lrwxrwxrwx  1 jere jere   19 2013-04-27 12:31 fig10.2 -> ../signals/sigusr.c
lrwxrwxrwx  1 jere jere   17 2013-04-27 12:31 fig10.20 -> ../signals/mask.c
lrwxrwxrwx  1 jere jere   21 2013-04-27 12:31 fig10.22 -> ../signals/suspend1.c
lrwxrwxrwx  1 jere jere   21 2013-04-27 12:31 fig10.23 -> ../signals/suspend2.c
lrwxrwxrwx  1 jere jere   17 2013-04-27 12:31 fig10.24 -> ../lib/tellwait.c
lrwxrwxrwx  1 jere jere   18 2013-04-27 12:31 fig10.25 -> ../signals/abort.c
lrwxrwxrwx  1 jere jere   21 2013-04-27 12:31 fig10.26 -> ../signals/systest2.c
lrwxrwxrwx  1 jere jere   19 2013-04-27 12:31 fig10.28 -> ../signals/system.c
```

Task 4:
Compiling Source Files in the Book

```
^_^jere@MBPR [~/SystemProgramming/.../figlinks]
^_^jere@MBPR [~/SystemProgramming/.../figlinks] ls -al fig1.3
lrwxrwxrwx 1 jere jere 14 2013-04-27 12:31 fig1.3 -> ../intro/ls1.c
```

```
^_^jere@MBPR [~/SystemProgramming/.../intro] ls
getcputc    hello    ls1    Makefile  mycat.c  shell1.c  shell2.c   testerror.c  uidgid.c
getcputc.c  hello.c  ls1.c  mycat     shell1   shell2    testerror  uidgid
^_^jere@MBPR [~/SystemProgramming/.../intro] rm ls1
ls1 deleted.
^_^jere@MBPR [~/SystemProgramming/.../intro] make ls1
gcc -ansi -I../include -Wall -DLINUX -D_GNU_SOURCE  ls1.c -o ls1  -L../lib -lapue
^_^jere@MBPR [~/SystemProgramming/.../intro] ls
getcputc    hello    ls1    Makefile  mycat.c  shell1.c  shell2.c   testerror.c  uidgid.c
getcputc.c  hello.c  ls1.c  mycat     shell1   shell2    testerror  uidgid
^_^jere@MBPR [~/SystemProgramming/.../intro] ./ls1 /home
..
.
jere
^_^jere@MBPR [~/SystemProgramming/.../intro]
```

Task 4:
Compiling Source Files in the Book

Figure 1.3. List all the files in a directory

```
#include "apue.h"
#include <dirent.h>

int
main(int argc, char *argv[])
{
    DIR             *dp;
    struct dirent   *dirp;

    if (argc != 2)
        err_quit("usage: ls directory_name");

    if ((dp = opendir(argv[1])) == NULL)
        err_sys("can't open %s", argv[1]);
    while ((dirp = readdir(dp)) != NULL)
        printf("%s\n", dirp->d_name);

    closedir(dp);
    exit(0);
}
```