

CS35L – *Winter 2019*

| | |
|---------------|--------------------------|
| Slide set: | 3.1 |
| Slide topics: | Diff, modifying programs |
| Assignment: | 3 |



How to Install Software

Windows

- Installshield
- Microsoft/Windows Installer

OS X

- Drag and drop from .dmg mount -> Applications folder

Linux

- rpm(Redhat Package Management)
 - RedHat Linux (.rpm)
- apt-get(Advanced Package Tool)
 - Debian Linux, Ubuntu Linux (.deb)
- **Good old build process**
 - **configure, make, make install**

Decompressing Files

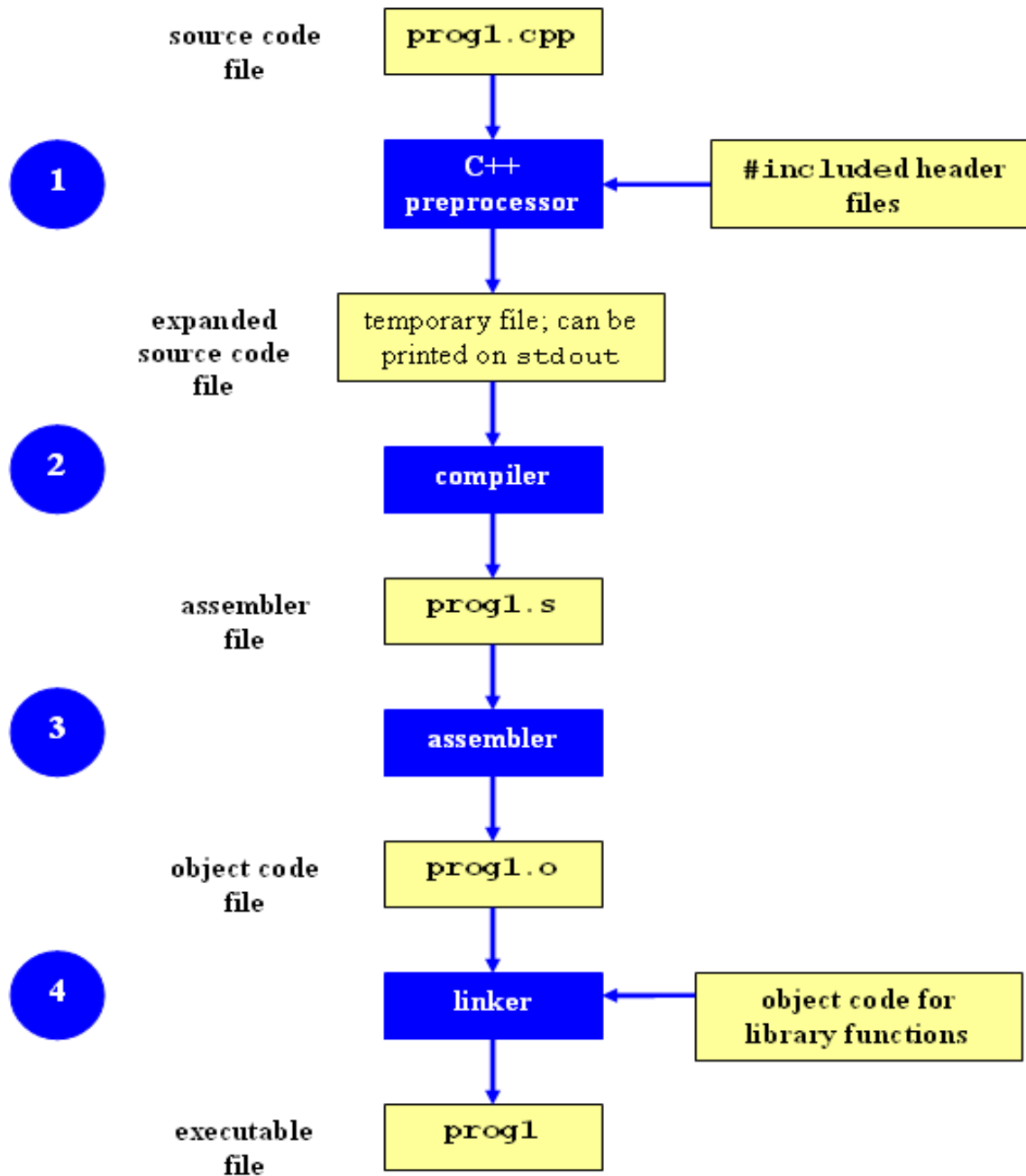


Generally, you receive Linux software in the tarball format (.tgz) or (.gz)

Decompress file in current directory:

- `$ tar -xzvf filename.tar.gz`
 - Option `-x`: `--extract`
 - Option `-z`: `--gzip`
 - Option `-v`: `--verbose`
 - Option `-f`: `--file`

COMPILATION PROCESS



Command-Line Compilation

main.cpp

- #includes cart.h

shop.cpp

- #includes lookup_item.h & cart.h

lookup_item.cpp

- #includes lookup_item.h

*Compilation
Command*

```
g++ -Wall lookup_item.cpp shop.cpp  
main.cpp -o shop
```



Changes to header or source files

- Rerun command to generate new executable
- But for only a small change?
(For example, a slight modification in `lookup_item.cpp`)
 - It isn't efficient to recompile `main.cpp` and `shop.cpp`
 - Solution: *avoid waste by producing a separate object code file for each source file*
 - `g++ -Wall -c lookup_item.cpp...`
(for each source file)
 - `g++ lookup_item.o main.o shop.o -o shop` (combine)
 - Less work for compiler, saves time but more commands

C++ Header Files

- C++ classes (and often function prototypes) are normally split up into two files.
- The header file has the extension of .h and contains class definitions and functions.
- The implementation of the class goes into the .cpp file.
- By doing this, if your class implementation doesn't change then it won't need to be recompiled.

C++ Header File Example

Compilation

- g++ main.cpp Num.cpp

File: Num.h

```
class Num {  
    private:  
        int num;  
    public:  
        Num(int n);  
        int getNum();  
};
```

File: Num.cpp

```
#include "Num.h"  
Num::Num() : num(0) { }  
Num::Num(int n): num(n) {}  
int Num::getNum() {  
    return num;  
}
```

File: main.cpp

```
#include <iostream>  
#include "Num.h"  
using namespace std;  
int main() {  
    Num n(35);  
    cout << n.getNum() << endl;  
    return 0;  
}
```


Changes to a header file

- Need to recompile every source file that includes it & every source file that includes a header that includes it.
 - In this example, changing `item.h` would affect `item.cpp` and `shop.cpp`
- Difficult to keep track of files when project is large
 - **Windows 7** ~40 million lines of code
 - **Google** ~2 billion lines of code



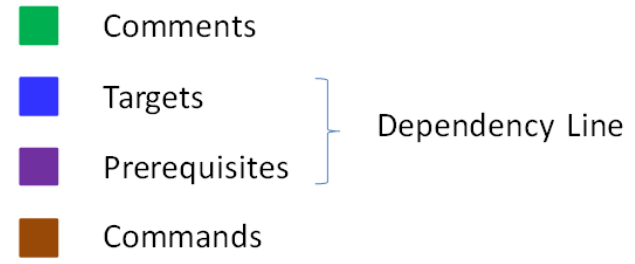
make

Utility for managing large software projects

Compiles files and keeps them up-to-date

Efficient Compilation (only files that need to be recompiled)

Makefile Example



```
# Makefile - A Basic Example
```

```
all : shop #usually first
```

```
shop : lookup_item.o main.o shop.o
```

```
    g++ -g -Wall -o shop item.o main.o shop.o
```

```
item.o : lookup_item.cpp lookup_item.h
```

```
    g++ -g -Wall -c lookup_item.cpp
```

```
main.o : shop.cpp cart.h
```

```
    g++ -g -Wall -c main.cpp
```

```
shop.o : shop.cpp lookup_item.h cart.h
```

```
    g++ -g -Wall -c shop.cpp
```

```
clean :
```

```
    rm -f item.o main.o shop.o
```

} Rule

Build Process

configure

- Script that checks details about the machine before installation
- Resolves dependencies between packages
- Creates 'Makefile'

make

- Requires 'Makefile' to run
- Compiles all the program code and creates executables in current temporary directory

make install

- make utility searches for a label named install within the Makefile, and executes only that section of it
- executables are copied into the final directories (system directories)

Patching



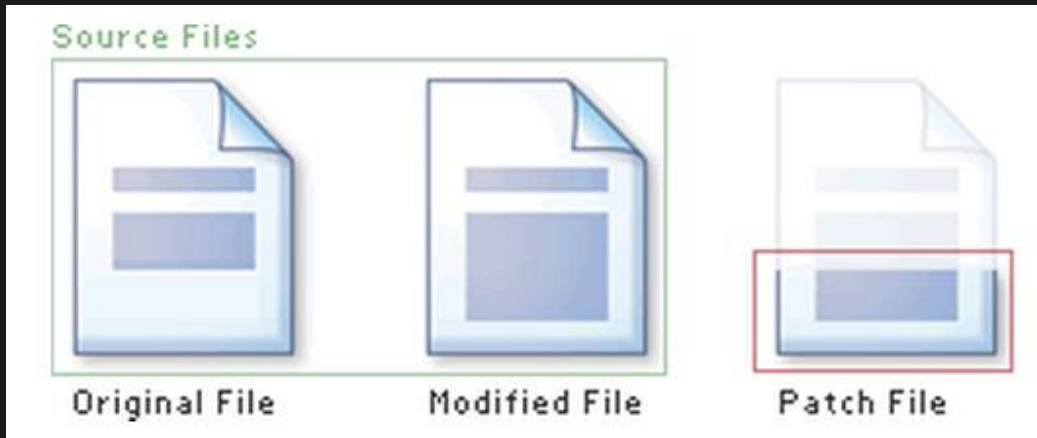
A patch is a piece of software designed to fix problems with or update a computer program



It's a diff file that includes the changes made to a file



A person who has the original (buggy) file can use the patch command with the diff file to add the changes to their original file



```
patch -pnum < patch_file
```

'man patch' to find
out what **pnum** does
and how to use it

APPLYING A PATCH

diff Unified Format

```
diff -u original_file modified_file
```

- --- path/to/original_file
- +++ path/to/modified_file
- @@ -l,s +l,s @@
 - @@: beginning of a hunk
 - l: beginning line number
 - s: number of lines the change hunk applies to for each file
- A line with a:
 - (- sign): was deleted from the original
 - (+ sign): was added to the original
 - (): stayed the same

Lab 3

- Coreutils 8.29 has a problem
 - `$ la -A` is equivalent to `ls -a -A`
 - if the current directory has two files named `.foo` and `bar`, the command `la -A` outputs four lines, one each for `.`, `..`, `.foo`, and `bar`.
 - These users want `la -A` to output just two lines instead, one for `.foo` and one for `bar`
- Why?
 - the `-a` option always overrides the `-A` option regardless of which option is given first
- Want the flag that comes later to take effect
- Fix the `ls` program

Getting Set Up (Step 1)

Download
coreutils-8.29 to
your home
directory

- Use 'wget'

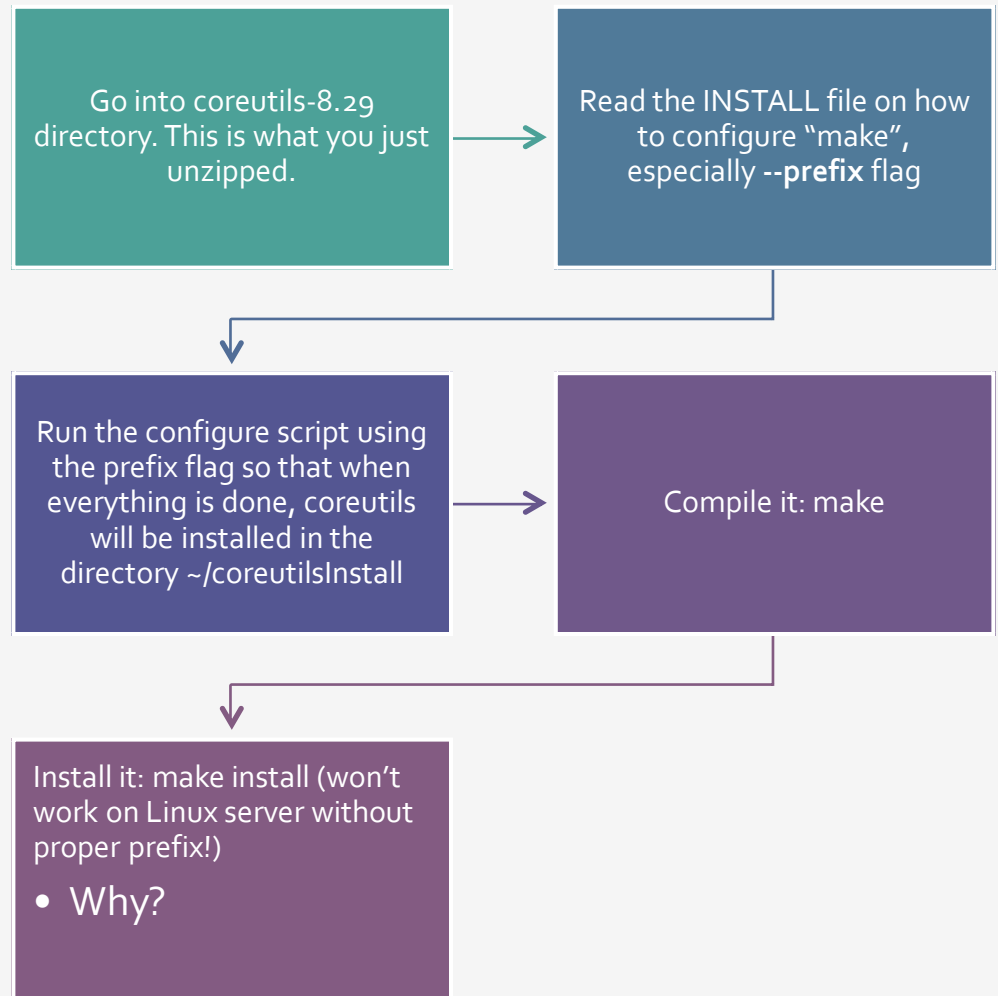
Untar and Unzip
it

- `tar -xJvf
coreutils-
8.29.tar.xz`

Make a directory
~/coreutilsInstall
in your home
directory (this is
where you'll be
installing
coreutils)


- `mkdir
~/coreutilsInstall`

Building coreutils (Step 2)



Reproduce Bug (Step 3)

Reproduce the bug by running the version of 'ls -a -A' in coreutils 8.29



If you just type \$ ls at CLI it won't run 'ls' in coreutils 8.29

Why? Shell looks for /bin/ls

To use coreutils 8.29: \$./ls

- This manually runs the executable in this directory

Patching and Building (Steps 4 & 5)

cd coreutils-8.29



vim or emacs
patch_file: copy and
paste the patch
content



patch -pnum <
patch_file

- 'man patch' to find out
what **pnum** does and
how to use it



cd into the coreutils-
8.29 directory and type
make to rebuild
patched ls.c.

- Don't install!

Testing Fix (Step 6)



Test the following:

Modified ls works

Installed unmodified ls
does NOT work



Test on:

Empty directory

Directory containing a
hidden file

- With just `-a`, with just `-A`
- With `-aA`
- With `-Aa`



Answer Q1 and Q2