Video Lecture # 05
Open-Source Softwares
Packaging with GNU autotools & cmake

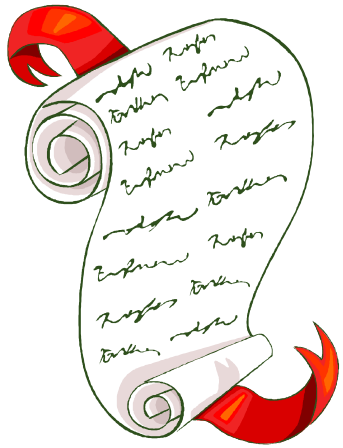
Course: SYSTEM PROGRAMMING

Instructor: Arif Butt

Punjab University College of Information Technology (PUCIT)
University of the Punjab

Today's Agenda

- Binary vs Open-Source s/w Packages
- Downloading and installing open-source softwares
- Packaging your software projects using
  - **`GNU autotools`**
  - **`Cmake`** utility



Binary Softwares
vs
Open-Source Softwares

Binary Software Packages

- A binary package is a collection of files bundled into a single file containing
  - executable files (compiled for a specific platform),
  - man/info pages,
  - copy right information,
  - configuration and installation scripts
- It is easy to install softwares from binary packages built for your machine and OS, as the dependencies are already resolved
- For the Debian based distributions (Ubuntu, Kali, Mint, ArchLinux) they come in **`.deb`** format and the package managers available are `apt`, `dpkg`, `aptitude`, `synaptic`
- For RedHat based distributions (Fedora, CentOS, OpenSuse) the packages come in **`.rpm`** format and the available package managers are `rpm` and `yum`.

Open-Source Software Packages

An Open-source software is a software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose (GNU GPL). Normally distributed as a tarball containing:

- Source code files
- README and INSTALL
- AUTHORS
- Configure script
- Makefile.am and Makefile.in

A source package is eventually converted into a binary package for a platform on which it is configured, build and installed. We normally use source packages to install softwares for following reasons:

- We cannot find a corresponding binary package
- We want to enhance functionalities of a software
- We want to fix a bug in a software

Downloading and Installing Open-Source Softwares

How to download OSS Packages?

- **Option 1:** You can download from some ftp repository using either your browser or may be the famous `wget` command from a Linux terminal.
- **Option 2:** You can use advanced packaging tool to download in present working directory by the following command:

```
$sudo apt-get source hello
```

- **Option 3:** You can also use **`git`** if the software is there on some public git repository like **`github.com`** or **`bitbucket.org`**

Magic Spell to Install Open-Source Packages

- A source package is eventually converted into a binary package for a platform on which it is `configured`, `build`, and `install`. Many a times, we all have recited the following magic spell to install a UNIX open-source tarball:

```
$ ./configure
$ make
$ sudo make install
```

Configure the Open-Source Software

```
$ ./configure
```

- The `configure` script makes sure that all of the dependencies for the rest of the `build` and `install` process are available. For example, for a software written in C, it ensures that the system have a C compiler, and find out what it is called and where to find it
- For our **`myexe`** package, once we execute the **`configure`** script it will create the **`Makefile`** that is required to build and install the software
- You can view the contents of the **`Makefile`** of this software, which contains about 3.4K lines

Build the Open-Source Software

$ make

- The build process runs a series of tasks defined in a `Makefile` to build the finished program from its source code. The tarball you download usually doesn't include a finished `Makefile`. Instead it comes with a template called `Makefile.in` and the `configure` script produces a customized `Makefile` specific to your system
- Once you give the **`make`** command, it will run the `Makefile` in the pwd or root directory of package, which may further call other makefiles (if any) in other directories and hence all source files in package will be compiled. This may take some time depending on your system and the size of the software

Install Open-Source Software

```
$ sudo make install
```

•The install process copies the build program and its libraries and documentation to the correct locations. The program's binary(ies) are copied to a directory on your `PATH`, and the program's manual page(s) are copied to a directory on your `MANPATH`. Depending on where the software is being installed, you might need escalated permissions to do this step

Un-installing Open-Source Software

- Makefile has many targets other than **install**, like **uninstall**, **clean** and **distclean**:

```
$sudo make uninstall
$hello
No such file or directory
$which hello
$man hello
No manual entry for hello
```

- The clean target can be used after the installation to remove all the object and executable files from the source directory.

```
$ make clean
```

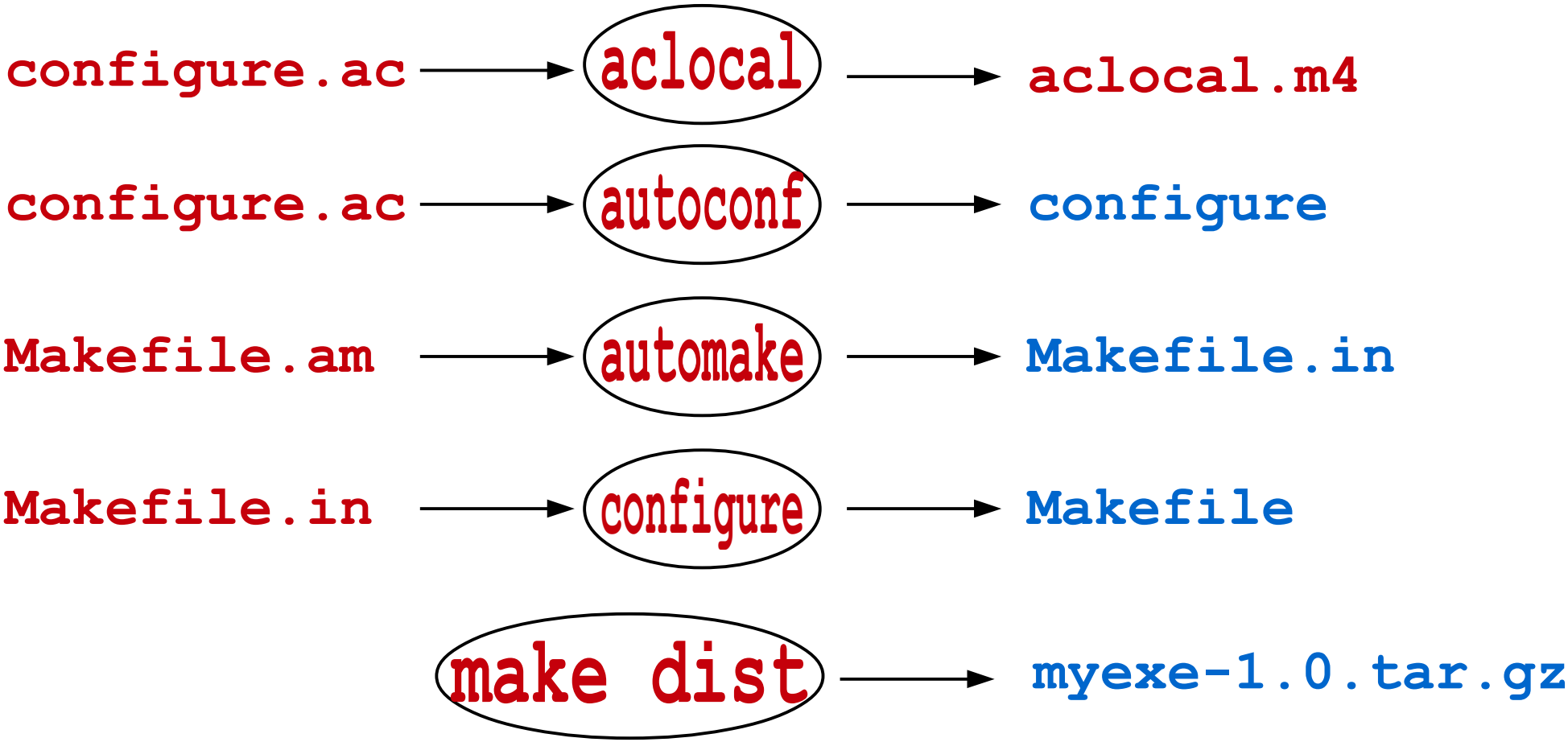
- Similarly, after installation if you want to remove all the files that were created by configure script.

```
$ make distclean
```

Packaging your software using GNU Autotools

autoconf & automake

Packaging s/w using GNU autotools



Packaging your software using cmake

What is cmake

In the simplest possible words, CMake is a cross platform Makefile generator. It is an effort to develop a better way to conf gure, build and deploy complex softwares written in various languages, across many different platforms like Linux, *UNICES, MacOS, MS Windows, iOS, Android,...

https://cmake.org

Friends of cmake

CMake has friends softwares that may be used on their own or together

- CMake: Build system generator
- CPack: Package generator used to create platform-specific installers
- CTest: A test driver tool used to run regression tests
- CDash: A web application for displaying test results and performing continuous integration testing

How Cmake work?

The `CMake` utility reads project description from a file named `CMakeLists.txt` and generates a Build System for a Makefile project, Visual Studio project, Eclipse project, Xcode project, …

Things To Do

If you have problems visit me in counseling hours. . . .