Using tar

An Introduction to `tar`

created by Brian Booth
modified by Stef Nychka

Department of Computing Science
University of Alberta

September 9, 2008

Overview

Introduction to Archiving

- What you submit often consists of more than one file.
- We want a way of packaging (*archiving*) our files together.
  - It is good etiquette to archive files if you are sending them to another person.
- You may have done this before.
  - e.g. WinZip (archives and compresses files)

`tar`: What it can do

`tar` (Tape ARchive) is a common program used to archive (package together) files. You can do the following (tar options, such as `c`, are in parentheses below):

- Create archive files (`c`)
- Extract files from an archive file (`x`)
- List the contents of an archive file (`t`)
- Use compression with any of the above (`z`)

Creating an Archive File

The $ is the command prompt. Lines without a $ are output from tar.

```
$ tar cvf submit.tar README Makefile hello.c hello.h
README
Makefile
hello.c
hello.h
$
```

- `c`: create the archive file
- `v`: the verbose option. Lists the files during creation
- `f submit.tar`: give the name `submit.tar` to the archive file
- `README Makefile hello.c hello.h`: Files to archive together

CAUTION

- Put name of tar file first, otherwise ...

```
$ tar cvf README Makefile hello.c hello.h
```

  ... `README` is overwritten, and becomes tar file containing `hello.c`, `Makefile`, and `hello.h`.
- Don't use a `-` (e.g. `-cfv`), otherwise what follows `f` is the tar file; so order of options becomes important.
- Always make a temporary directory, copy files to be archived there, and then `tar`. This way, you won't overwrite your original files accidentally.

Example of Archiving

Follow along with your TA. First, get sample tarred and compressed file (will follow similar procedures throughout the course):

1. `mkdir tar_eg`, `cd tar_eg`
2. Download `files.tar.gz` from top of home page, beneath link to this presentation
3. `tar xvfz files.tar.gz`, `ls -l`, `ls orig/`

Now, copy files to a temporary directory, and `tar` them:

1. `mkdir tar_tmp`, `cd orig`, copy files to `tar_tmp`
2. `cd ../tar_tmp`
3.

```
tar cvf submit.tar Makefile README hello.h hello.c
```

Listing the contents of an Archive File

```
$ tar tf submit.tar
README
Makefile
hello.c
hello.h

$ tar tvf submit.tar
-rw------- stef/csuser 306 2007-07-20 19:56 Makefile
-rw------- stef/csuser 163 2007-07-20 19:53 README
-rw------- stef/csuser 199 2007-07-20 19:57 hello.c
-rw------- stef/csuser 129 2007-07-20 19:58 hello.h
$
```

Extracting Files from an Archive File

```
$ tar xvf submit.tar
README
Makefile
hello.c
hello.h
$
```

After submitting, test that your tar file contains the proper files: make a temporary directory, do tar extract there so existing files aren't overwritten.

Example of Extraction

Follow along with your TA:

1. `cd` back to `tar_eg`
2. Make a directory called `untar_tmp`
3. Copy `submit.tar` from `tar_tmp` to `untar_tmp`
4. `cd` into `untar_tmp`
5. Extract files from `submit.tar` using `xvf`

Can edit files and use `ls` to ensure files are correct. Do this after submitting, and if find problems, fix them, move `submit.tar` to a permanent location, and submit again.

Summary of Commands

- Creating an archive file
  - `tar cvf name_here.tar file1 ...`
  - always use a copy and a tmp directory
- Listing contents of an archive file
  - `tar tvf name_here.tar`
- Extracting files from an archive file
  - `tar xvf name_here.tar`
  - always use a copy and a tmp directory