

Fall 2005 Handout 10

Archive several files into one big .tar file

tar is a utility for writing a group of files onto a tape, creating a *tape archive*. But the output of **tar**, like that of any Unix program, can be directed to a file instead of to a hardware device. Give the file a name ending with **.tar**.

The following example could have used a device name such as **/dev/rmtnh** (raw magnetic tape) instead of the filename **date.tar**. In that case you'd also need the **mt rewind** command.

```
1$ cd
2$ date > date1
3$ date > date2
4$ date > date3
5$ ls -l date1 date2 date3           or use wildcard in p. 28; ksh(1) p. 14: ls -l date[1-3]
-rw-----  1 mm64      users          29 Dec 15 14:24 date1
-rw-----  1 mm64      users          29 Dec 15 14:24 date2
-rw-----  1 mm64      users          29 Dec 15 14:24 date3

6$ tar cvf date.tar date[1-3]       create date.tar
a date1 1K
a date2 1K
a date3 1K

7$ ls -l date.tar
-rw-----  1 mm64      users          4096 Dec 15 14:24 date.tar

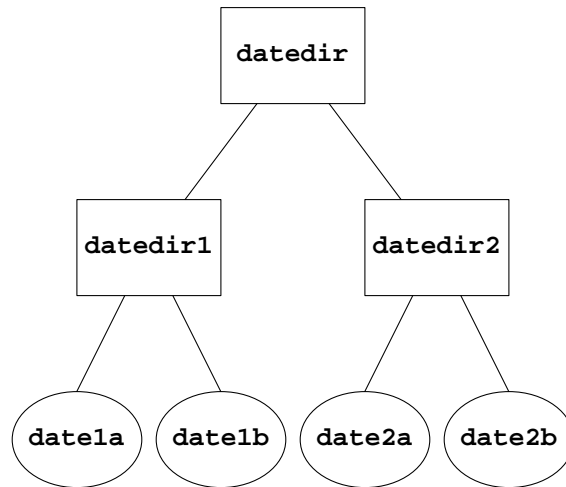
8$ tar tvf date.tar | more           Output a table of contents of the .tar file.
tar: blocksize = 8
-rw----- 50766/15      29 Dec 15 14:24 2005 date1
-rw----- 50766/15      29 Dec 15 14:24 2005 date2
-rw----- 50766/15      29 Dec 15 14:24 2005 date3
```

The **x** flag extracts a copy of one or more of the little files archived in the **.tar** file. To extract everything archived in the **.tar** file, simply give no arguments after the name of the **.tar** file. The files you recreate will be owned by their original owners if you are the superuser. Extraction does not change the contents of the **tar** file.

```
9$ rm date[1-3]

10$ tar xvf date.tar date1          create date1
tar: blocksize = 8
x date1, 29 bytes, 1 tape blocks

11$ ls -l date1
-rw-----  1 mm64      users          29 Dec 15 14:24 date1
```

Archive an entire directory into one big .tar file

If you give one or more directory names instead of one or more filenames to **tar cvf** after the name of the **.tar** file, **tar** will archive the directories and all of their descendants, including all of the files they contain.

```
1$ cd
2$ pwd
/home1/a/abc1234

3$ mkdir datedir

4$ mkdir datedir/datedir1
5$ date > datedir/datedir1/date1a
6$ date > datedir/datedir1/date1b

7$ mkdir datedir/datedir2
8$ date > datedir/datedir2/date2a
9$ date > datedir/datedir2/date2b

10$ tar cvf date.tar datedir
a datedir/ OK
a datedir/datedir1/ OK
a datedir/datedir1/date1a 1K
a datedir/datedir1/date1b 1K
a datedir/datedir2/ OK
a datedir/datedir2/date2a 1K
a datedir/datedir2/date2b 1K
```

```

11$ tar tvf date.tar | more
tar: blocksize = 13
drwx--x--x 50766/15      0 Dec 15 14:24 2005 datedir/
drwx--x--x 50766/15      0 Dec 15 14:24 2005 datedir/datedir1/
-rw----- 50766/15     29 Dec 15 14:24 2005 datedir/datedir1/date1a
-rw----- 50766/15     29 Dec 15 14:24 2005 datedir/datedir1/date1b
drwx--x--x 50766/15      0 Dec 15 14:24 2005 datedir/datedir2/
-rw----- 50766/15     29 Dec 15 14:24 2005 datedir/datedir2/date2a
-rw----- 50766/15     29 Dec 15 14:24 2005 datedir/datedir2/date2b

12$ rm    datedir/datedir[12]/*
13$ rmdir datedir/datedir[12]
14$ rmdir datedir

```

The following `tar xvf` command will re-create the directory `datedir`, its subdirectory `datedir2`, and the file `date2a`:

```

15$ tar xvf date.tar datedir/datedir2/date2a
tar: blocksize = 13
x datedir/datedir2/date2a, 29 bytes, 1 tape blocks

16$ ls -l | more
drwx--x--x   3 mm64      users          182 Dec 15 14:24 datedir

17$ ls -l datedir | more
drwx--x--x   2 mm64      users          180 Dec 15 14:24 datedir/datedir2

18$ ls -l datedir/datedir2/date2a
-rw-----   1 mm64      users           29 Dec 15 14:24 datedir/datedir2/date2a

```

Include the full pathname in the .tar file

If you specify the full pathnames of the files and directories to be archived, then their full pathnames will be stored in the `.tar` file:

```

1$ rm date.tar
2$ tar cvf date.tar ~/datedir
a /home1/a/abc1234/datedir/ OK
a /home1/a/abc1234/datedir/datedir2/ OK
a /home1/a/abc1234/datedir/datedir2/date2a 1K
a /home1/a/abc1234/datedir/datedir2/date2b 1K
a /home1/a/abc1234/datedir/datedir1/ OK
a /home1/a/abc1234/datedir/datedir1/date1a 1K
a /home1/a/abc1234/datedir/datedir1/date1b 1K

3$ tar tvf date.tar
tar: blocksize = 13
drwx--x--x 50766/15      0 Dec 15 14:24 2005 /home1/a/abc1234/datedir/
drwx--x--x 50766/15      0 Dec 15 14:24 2005 /home1/a/abc1234/datedir/datedir2/
-rw----- 50766/15     29 Dec 15 14:24 2005 /home1/a/abc1234/datedir/datedir2/date2a
-rw----- 50766/15     29 Dec 15 14:24 2005 /home1/a/abc1234/datedir/datedir2/date2b
drwx--x--x 50766/15      0 Dec 15 14:24 2005 /home1/a/abc1234/datedir/datedir1/
-rw----- 50766/15     29 Dec 15 14:24 2005 /home1/a/abc1234/datedir/datedir1/date1a
-rw----- 50766/15     29 Dec 15 14:24 2005 /home1/a/abc1234/datedir/datedir1/date1b

```

If you specify only the basenames of the files and directories to be archived, then only their base-names will be stored in the `.tar` file:

```
4$ rm date.tar
5$ cd ~/datedir/datedir1
6$ tar cvf ~/date.tar datela datelb
a datela 1K
a datelb 1K

7$ cd ~/datedir/datedir2
8$ tar rvf ~/date.tar date2a date2b           "replace", not cvf!
a date2a 1K
a date2b 1K

9$ tar tvf ~/date.tar
tar: blocksize = 10
-rw----- 50766/15      29 Dec 15 14:24 2005 datela
-rw----- 50766/15      29 Dec 15 14:24 2005 datelb
-rw----- 50766/15      29 Dec 15 14:24 2005 date2a
-rw----- 50766/15      29 Dec 15 14:24 2005 date2b
```

Another way to do exactly the same thing is to use the uppercase `-C` option of `tar` instead of doing the `cd`'ing yourself:

```
10$ cd
11$ rm date.tar
12$ tar cvf date.tar \
    -C datedir/datedir1 datela \
    -C datedir/datedir1 datelb \
    -C datedir/datedir2 date2a \
    -C datedir/datedir2 date2b
a datela 1K
a datelb 1K
a date2a 1K
a date2b 1K

13$ pwd
/home1/a/abc1234           The -C effected only the tar, not you.

14$ tar tvf date.tar
tar: blocksize = 10
-rw----- 50766/15      29 Dec 15 14:24 2005 datela
-rw----- 50766/15      29 Dec 15 14:24 2005 datelb
-rw----- 50766/15      29 Dec 15 14:24 2005 date2a
-rw----- 50766/15      29 Dec 15 14:24 2005 date2b
```

Exclude one or more files and/or subdirectories

To archive the directory `datedir` and all of its descendants except for the directory `datedir/datedir1`, use the `-e` option. Specify the exception directory `datedir/datedir1` *before* its ancestor directory `datedir`:

```
1$ cd
2$ tar cvf date.tar -e datedir/datedir1 datedir
```

```
3$ tar tvf date.tar
```

▼ Homework 10.1: create a tar file

Verify that all of the above works. If you're allowed to use a tape drive, **tar** some files to tape instead of to a **.tar** file.

▲

▼ Homework 10.2: copy all your files from one machine to another

```
1$ cd
2$ pwd

3$ tar cvf all.tar .
4$ tar tvf all.tar | more
5$ tar tvf all.tar | lpr
```

Use **sftp** to copy the above **binary** file named **all.tar** to your home directory on another machine. Then on the other machine,

```
$ cd
$ pwd

$ ls -l all.tar
$ tar tvf all.tar | more
$ tar tvf all.tar | lpr
$ tar xvpf all.tar      p 'cause hosts have different umask (Hand 2, pp. 13-14, ll. 75-78)
```

▲

Compress and uncompress

```
1$ cd
2$ cp /etc/passwd .           Copy /etc/passwd to your current directory.
3$ ls -l passwd
-r-----  1 abc1234  users      123343 Dec 15 14:24 passwd

4$ compress passwd          remove passwd and create passwd.Z
5$ ls -l passwd.Z
-r-----  1 abc1234  users      45720 Dec 15 14:24 passwd.Z
```

Do not **cat** to the screen or **lpr** a compressed file (in this case, a **.Z** file). Instead,

```
6$ zcat passwd.Z | head -3
root:x:0:1:Super-User:/root:/sbin/sh
daemon:x:1:1:/:
bin:x:2:2:/:usr/bin:
```

```
7$ uncompress passwd.Z      remove passwd.Z and create passwd
8$ ls -l passwd
-r-----  1 abc1234  users      123343 Dec 15 14:24 passwd
```

The compression and decompression is not “lossy”:

```
9$ cmp passwd /etc/passwd   No output if identical: Handout 2, p. 12.
10$ rm passwd
```

Other compression programs

<i>suffix</i>	<i>compress</i>	<i>decompress</i>
.Z	compress	uncompress
.gz	gzip	gunzip
.z	pack	unpack

Programs written by GNU often start with **g**. See <http://www.gnu.org/>. To list the pairs of utilities for compression and decompression on i5.nyu.edu,

```
1$ man -k compress | more
```

search for a keyword with -k

David Lemson used to maintain a comprehensive list of suffixes, pairs of utilities, and how to get them.

<http://www.lemson.com/lemson/work-uiuc.html>

▼ Homework 10.3: compress and uncompress a file

Compress and uncompress a file (ASCII or binary). Use **bc** to find the percent by which it was compressed. Which pair of utilities yields the most compression? Do text files compress further than image files?

```
1$ bc
scale = 2
100 * 45720 / 123343
37.06
control-d
2$
```

*Handout 2, p. 21; Handout 3, p. 11; Handout 4, pp. 23–24; Handout 7, p. 7
Output answers to two decimal places.*

▲

Watch your disk usage soar!

```
1$ du ~ | more
2$ du -s ~
385350 /home1/m/mm64
```

*tilde is your home directory; unit is ¹/₂ K
sum: only one line of output*

If you run out of disk space

List all your files in decreasing size order. The **+6nr** tells **sort** to ignore the first six fields on each line. It will then sort the lines in reverse numeric (i.e., decreasing numeric) order of the seventh field.

```
1$ find ~ -type f -ls | sort +6nr | more
641867 6032 -rwx----- 1 mm64 users 6161984 Mar 9 2003 /home1/m/mm64/public_html/x52.9547/sr
641298 5232 -r--r--r-- 1 mm64 users 5344181 Sep 17 1998 /home1/m/mm64/public_html/x52.9545/sr
641747 4728 -rwx----- 1 mm64 users 4830592 Mar 9 2003 /home1/m/mm64/public_html/x52.9547/sr
641799 4656 -rwx----- 1 mm64 users 4754916 Mar 9 2003 /home1/m/mm64/public_html/x52.9547/sr
641880 3928 -rw----- 1 mm64 users 4014080 Mar 9 2003 /home1/m/mm64/public_html/x52.9547/sr
```

The tilde stands for the full pathname of your home directory. The **-type f** lists only the files, not the directories. Remove the files that you no longer need, and try again.

Save yourself typing

```
1$ v=httpd-2.0.39
2$ echo $v
```

version number

```
3$ lynx -source http://httpd.apache.org/dist/httpd/$v.tar.gz > $v.tar.gz
```

Download, compile, and configure the Apache Web Server

(1) To download the Apache web server to i5.nyu.edu, **lynx** (“links”) is easier than **ftp**, and is easier than **lynx**.

```
1$ cd
2$ lynx -source \
  http://httpd.apache.org/dist/httpd/httpd-2.0.39.tar.gz > \
  httpd-2.0.39.tar.gz

3$ ls -l httpd-2.0.39.tar.gz
-rw----- 1 mm64 users 4413542 Jun 25 22:15 httpd-2.0.39.tar.gz
```

The above **lynx** (“links”) is easier than **ftp**. But if you have to use **ftp**, see the list of **ftp** sites at <http://www.apache.org/dist/>.

(2) To decompress the server,

```
4$ gunzip httpd-2.0.39.tar.gz
5$ ls -l httpd-2.0.39.tar
-rw----- 1 mm64 users 20152320 Jun 25 22:15 httpd-2.0.39.tar
```

```
6$ bc
scale = 2
100 * 4413542 / 20152320
21.90
control-d
```

(3) To verify that it was downloaded and decompressed correctly, ask for a verbose table of contents **tv**. The **f** means that the following argument is the name of the **.tar** file.

```
7$ tar tvf httpd-2.0.39.tar | more
drwxr-xr-x 0/0 0 Jun 17 23:46 2002 httpd-2.0.39/
drwxr-xr-x 0/0 0 Jun 17 23:46 2002 httpd-2.0.39/build/
drwxr-xr-x 0/0 0 Jun 17 23:46 2002 httpd-2.0.39/build/win32/
-rw-r--r-- 0/0 1078 Sep 3 12:43 2001 httpd-2.0.39/build/win32/apache.ico
-rw-r--r-- 0/0 2997 Mar 13 15:47 2002 httpd-2.0.39/build/win32/win32ver.awk
```

```
8$ tar tvf httpd-2.0.39.tar | wc -l
1996
```

```
9$ tar tvf httpd-2.0.39.tar | grep README | more
-rw-r--r-- 0/0 3306 Apr 23 08:18 2002 httpd-2.0.39/README
-rw-r--r-- 0/0 1854 May 6 01:25 2002 httpd-2.0.39/README.platforms
-rw-r--r-- 0/0 1238 Mar 13 15:47 2002 httpd-2.0.39/docs/error/README
-rw-r--r-- 0/0 4872 Aug 24 01:33 1999 httpd-2.0.39/docs/icons/README
-rw-r--r-- 0/0 300 Aug 24 01:34 1999 httpd-2.0.39/docs/icons/small/README.txt
```

(4) To “extract” the files you want,

```
10$ tar xvf httpd-2.0.39.tar httpd-2.0.39/README httpd-2.0.39/INSTALL
x httpd-2.0.39/INSTALL, 3741 bytes, 8 tape blocks
x httpd-2.0.39/README, 3306 bytes, 7 tape blocks
```

The above **tar xvf** will create a **httpd-2.0.39** subdirectory of your current directory. (Your current directory is currently your home directory.)

```
11$ ls -ld httpd-2.0.39           List the directory itself.
12$ ls -l httpd-2.0.39           List the two files in the directory.
```

(5) When you “extract” all of the 1,996 directories and files, save the output in a file in case you need to look at it later. `tar xvf` will create the `httpd-2.0.39` subdirectory if you have not already created it.

```
13$ du -s ~
14$ tar xvf httpd-2.0.39.tar > tar.out
15$ wc -l tar.out                 Should contain 1,996 lines.
16$ du -s ~

17$ ls -ld httpd-2.0.39
18$ ls -l httpd-2.0.39 | more

19$ du -s ~
20$ rm httpd-2.0.39.tar tar.out   Now that you've extracted everything.
21$ du -s ~
```

If you didn't have enough disk space, `tar xvf` will give you a warning like

```
msgcnt 29 vxfs: msg 044: vx_bsdquotaupdate - warning:
/home1 file system user 50766 disk quota exceeded
tar: httpd-2.0.39/src/lib/apr-util/xml/expat/aclocal.m4:
HELP - extract write error
tar: close error: Disc quota exceeded
```

In this case, remove all the directories and files that `tar xvf` just created:

```
22$ rm -R httpd-2.0.39           “recursive”: visit all directories below
```

(6) Create the directory that will hold the web server:

```
23$ cd
24$ mkdir apache
```

(7) Configure the web server. The dot slash makes us run the `configure` in the current directory. We have to use the `$HOME` environment variable because a tilde can be used only at the start of an argument.

```
25$ cd httpd-2.0.39
26$ ls -l configure
27$ ./configure -help | more
28$ ./configure --prefix=$HOME/apache   option starts with two dashes
```

You'll see a long list of `checking for`'s.

(8) Compile the web server:

```
29$ du -s ~
30$ ls -l Makefile
31$ make
32$ du -s ~
```

If it stops with an error, it may be because you ran out of disk space (again). Remove some files and give the above `make` command again.

(9) Install the web server:

```
33$ make install
```

If you get an error message like


```
cp: /home1/m/mm64/apache/icons/small/rainbow.png: Disc quota exceeded
```

remove more files and **make install** again.

(10) Configure the web server:

```
34$ cd ~/apache/conf
35$ awk '$1 == "Listen"' httpd.conf
Listen 80
```

Edit **httpd.conf** and change the port number to a number in the range 1025–65535. For example, use the last five digits of your social security number.

(11) Start the web server:

```
36$ cd ~/apache/bin
37$ apachectl start
38$ ps -Af | grep abc1234 | more           quite a swarm of them
39$ netstat -a -f inet -P tcp | grep 12345 last 5 digits of your social security number
```

Now point a web browser at **http://i5.nyu.edu:12345/**, where **12345** is the port number you specified.

Place your web pages in the **~/apache/htdocs** directory and its subdirectories. Name your home page **index.html**.

The simplest Java applet

An *applet* is a little program embedded in a web page. Create a file named **MyApplet.java** in your **~/public_html** directory:

—On the Web at

http://i5.nyu.edu/~mm64/x52.9547/src/MyApplet.java

```
1 import java.applet.*;    //Applet Package
2 import java.awt.*;      //Abstract Windowing Toolkit Package
3
4 public class MyApplet extends Applet {
5     public void paint(Graphics g)
6     {
7         g.setColor(Color.blue);
8         g.drawRect(0, 0, 20, 10);    //upper left corner, lower right
9         g.drawString("lower left", 0, 99); //will cut off descenders
10
11         g.setColor(Color.red);
12         g.drawLine(149, 0, 149, 99); //down the right edge
13     }
14 }
```

```
1$ javac -verbose MyApplet.java      Create MyApplet.class.
```

```
2$ ls -l MyApplet.class
```

```
3$ chmod 444 MyApplet.class        Turn on all three r's :r--r--r--
```

```
4$ ls -l MyApplet.class
```

Write the following in your web page **~/public_html/index.html**:

```
<APPLET CODE = "MyApplet.class"  
WIDTH = 150  
HEIGHT = 100  
ALT = "This browser understands the APPLET tag but does not have Java enabled now.">  
This browser does not understand the APPLET tag and does not support Java.  
</APPLET>
```

The World Wide Web Consortium wants you to write

```
<OBJECT CODETYPE = "application/java"  
CLASSID = "java:MyApplet.class"  
WIDTH = 150  
HEIGHT = 100>  
This browser does not understand the OBJECT tag and does not support Java.  
</OBJECT>
```

~

□