Some other Tools in UNIX

Prof. P.C.P. Bhatt

Compression Using tar - 1

As such *tar,* by itself, preserves the ASCII code and does not compress information. Unix provides a set of compression utilities which include a *compress* and a *uuencode* command. The command structure for the *compress* or *uncompress* command is as follows:

compress *options filename*

uncompress *options filename*

On executing the *compress* command we will get file with a .Z extension, i.e. with a file filename we get filename.Z file. Upon executing *uncompress* command with filename.Z as argument, we shall recover the original file filename.

Compression Using tar - 1

The example below shows a use of *compress* (also *uncompress*)
command which results in a .Z file.

```
bhatt@SE-0 [T] >>cp cfiles.tar test; compress test; ls
M ReadMe cfiles.tar test.Z
bhatt@SE-0 [T] >>uncompress test.Z; ls
M ReadMe cfiles.tar test
```

Compression Using tar - 2

Another method of compression is to use the *uuencode* command. It is quite common to use a phrase like *uuencode* a file and then subsequently use *uudecode* to get the original file. Let us *uuencode* our test file. The example is shown below:

```
bhatt@SE-0 [T] >>uuencode test test > test.uu ; ls; rm test ;
\ls ; uudecode test.uu ; rm test.uu; ls
    ReadMe    cfiles.tar    test test.uu M
    ReadMe    cfiles.tar    test.uu M
    ReadMe    cfiles.tar    test
```

The way to use *uuencode/uudecode* is as follows:

```
uuencode my_tar.tar my_tar.tar > my_tar.uu
```

Zip and Unzip

Various Unix flavours, as also MS environments, provide Instructions to compress a file with the *zip* command. A compressed file may be later unzipped by using an *unzip* command. In GNU environment the corresponding commands are *gzip* (to compress) and *gunzip* (to uncompress).

Below is a simple example which shows use of these commands:

```
bhatt@SE-0 [T] >>gzip test; ls; gunzip test.gz; ls;
M ReadMe cfiles.tar test.gz
M ReadMe cfiles.tar test
```

FTP - 1

Network file transfers: The most frequent mode of file transfers over the net is by using the file transfer protocol or FTP. To perform file transfer from a host we use the following command.

```
ftp <host-name>
```

This command may be replaced by using an *open* command to establish a connection with the host for file transfer. One may first give the *ftp* command followed by *open* as shown below :

```
ftp
open <host-name>
```

FTP - 2

With anonymous or guest logins, it is a good idea to input one's e-mail contact address as the password. A short prompt may be used sometimes to prompt the user. Below we show an example usage:

user anonymous e-mail-address

Binary files must be downloaded using the BINARY command. ASCII files too can be downloaded with binary mode enabled. FTP starts in ASCII by default. Most Commonly used *ftp* commands are *get* and *put*. See the example usage of the *get* command.

get <rfile> <lfile>

This command gets the remote file named *rfile* and assigns it a local file name lfile.

FTP - 3

Within the FTP protocol, the *hash* command helps to see the progression of the *ftp* transfers. This is because of # displayed for every block transfer (uploaded or downloaded). A typical get command is shown below.

```
ftp> hash
ftp> binary
ftp> get someFileName
```

Example - 1

In the example below, we additionally use compression on the tarred files.

1. Make a tar file: create xxx.tar file
2. Compress: generate xxx.tar.z file
3. Issue the ftp command: ftp

Below we have an example of such a usage:

Step 1: $ tar -cf *graphics.tar /pub/graphics*

This step takes all files in /pub/graphics and its subdirectories and creates a tar fille named graphics.tar.

Step 2: $ compress *graphics.tar*

This step will create graphics.tar.z file.

Step 3: uncompress graphics.tar.z to get graphics.tar

Step 4: *tar gf graphics.tar* will give file gf .

Image File Formats for Internet Applications

.AVI	Audio visual interleave
.DL	Animated picture files
.PCX	IBM PC image file
.WPG	A word perfect file
.BMP	Bitmap file
.RAW	24 bit RGB picture file.

Example - 2

File sizes up to 100{300K are not uncommon for .gif files. Often UseNet files are delimited to 64k. A typical 640*480 VGA image may require transmission in multiple parts. Typical .gif file in Unix environment begins as follows:

begin 640 image.gif (640 represents the access rights of file.)

Steps for getting a .gif or .jpg files may be as follows:

Step 1: Get all the parts of the image file as part files.

Step 2: Strip mail headers for each part - file.

Step 3: Concatenate all the parts to make one .uue file.

Step 4: *uudecode* to get a .gif/.jpg or .zip file.

Step 5: If it is a .zipfile unzip it.

Step 6: If it is a .jpgfile either use jpg image viewer like cview or, alternatively, use jpg2gif utility to get .gif file.

Step 7: View image from .gif file.

Performance Analysis and Profiling

Profiler - gives an estimate of the time spent in each of the functions.

Text processing (improving performance)

strlen – returns the length of the string
strcpy – copy strings
memcpy – memory block copy

Steps To Analyse Performance of *c* Programs:

The following steps will walk us through the basic steps :

1. Compile with p option, the profiler option

   cc -p a.c   (Additionally, use -o option for linked routines.)

2. Now run the program *a.out.* (This step results in a mon.out file in the directory)

3. Next see the profile by using prof command as follows:

   prof a.out

Profiling : Example

First let us study the following program in *c*:

```
#include <stdio.h>
#include <ctype.h>
int a1;
int a2;
add() /* adds two integers */
{
int x;
int i;
for (i=1; i <=100000; i++)
x = a1 + a2;
return x;
}
main()
{
int k;
a1 = 5;
a2 = 2;
for (k=1; k <=1000000; k++);;
printf("The addition gives %d \n", add());
}
```

Now let us see the way it has been pro¯led.

```
bhatt@SE-0 [P] >>cc -p a.c
bhatt@SE-0 [P] >>a.out
The addition gives 7
bhatt@SE-0 [P] >>ls
ReadMe a.c a.out mon.out
bhatt@SE-0 [P] >>prof a.out
%Time  Seconds   Cumsecs  #Calls   msec/call   Name
77.8     0.07      0.07      1       70.        main
22.2     0.02      0.09      1       20.        add
bhatt@SE-0 [P] >>
```