

Random Bytes – June 2008

Setting up a stand-alone /home partition.

If you are updating an existing installation to set up a separate /home partition, you will most likely want to use an extra hard drive to allocate more disk space for your home directory. The routine for this is something like ..

Install new HDD if required, and, using a Live CD with a suitable partitioning tool, create a new partition. Then create a file system on the new partition using something like ..

```
# mkfs -t ext3 /dev/hdxx
```

where xx is the partition id
eg/. hda3

Start up a root terminal and create a mount point for temporary mounting of devices, eg/. /mnt/tmp

```
# mkdir /mnt/tmp
```

Mount the new drive on the temporary mount point

```
# mount /dev/hdxx /mnt/tmp
```

Random Bytes – June 2008

Use the tar command to copy all the existing /home directory tree to the new drive.

```
# cd /home  
# tar cf - . | (cd /mnt/tmp; tar xvf - )
```

Note that the first tar command creates an archive, and writes to standard output. The second tar command extracts the tar archive which is being piped to its standard input.

Also note that the tar program saves the ownership and permissions of files in the archive, as well as saving symbolic and hard links. It also preserves the full directory structure.

Using a similar command, make another copy to a safe place, eg/. A USB drive, as a safety backup. For example, if you have a USB drive mounted on /media you could use something like ..

```
# tar cf - . | (cd /media/usb-disk/backup; tar xvf - )
```

Random Bytes – June 2008

Unmount the newly created /home partition from /mnt/tmp

```
# umount /mnt/tmp
```

note: *umount* not *unmount*

Now we need to delete the contents of the original /home directory to free up some disk space.

```
# cd /home  
# pwd  
# rm -rf *
```

Finally, you will need to add a new line to the /etc/fstab file, to tell the system that there is a new /home partition, and where to mount it.

Using your favourite editor, add a line to /etc/fstab as follows, for example ..

```
/dev/hda3          /home      ext3        defaults    0          2
```

```
<file system> <mount point> <fs type> <options> <dump> <pass>
```

Reboot into your newly modified installation, and you should have extra space in your root partition, and a new home partition, also with extra space.

Checking end of line in text files.

PC Update, like many club magazines and newsletters, invite submissions in plain text format. Those of us using Linux to prepare these files need to be aware that text files in Linux use only a line feed to signify end-of-line. Windows files typically use the MSDOS convention, namely carriage-return + line feed.

We are able to check which end-of-line convention is in the text file by using the `cat` and `od` commands, or using a GUI patch editor.

Start up your favourite text editor, and create a simple 3 line file.

Start up a terminal, and use the `cat` command to display the non-printing characters, for example ..

```
$ cat -v test-gedit.txt
```

where `-v` is the option to display non-printing characters

Random Bytes – June 2008

```
$ cat -v test-gedit.txt  
This is a test file.
```

This is the second line

If we use the same command on a text file with MSDOS type end of line markers, we see something like ..

```
$ cat -v text-crlf.txt  
This is a new file with crlf enabled^M  
^M  
This is the second line.^M
```

Similarly, for a Mac style text file ..

```
$ cat -v test.mac.txt  
This is a test file.^M^MThis is the second line.^M$
```

Random Bytes – June 2008

You will, depending on the editor setup, see the lines of text with non printing characters displayed as control codes, eg/. ^M.

However, if only line feeds are used to indicate end-of-line, `cat -v` won't display the end-of-line characters.

In this case, you will need to use the `od` command.

The command

```
$ od -c test.txt
```

will give an ascii dump of the file on the display. For example

```
0000000 T h i s   i s   a   t e s t   f
0000020 i l e . \n \n T h i s   i s   t h
0000040 e   s e c o n d   l i n e \n \n
0000057
```

Random Bytes – June 2008

If we look at a text file formatted as a MSDOS file with **od**, we see ..

```
0000000 T h i s   i s   a   n   e   w   f   i
0000020 l e   w   i   t   h   c   r   l   f   e   n   a
0000040 b   l   e   d   \r \n \r \n T h i s   i   s
0000060 t   h   e   s   e   c   o   n   d   l   i   n   e   .
0000100 \r \n
0000102
```

Here, `\r` denotes a 'carriage return'.

A file formatted for a Mac looks like this ..

```
0000000 T h i s   i s   a   t   e   s   t   f
0000020 i   l   e   . \r \r T h i s   i   s   t   h
0000040 e   s   e   c   o   n   d   l   i   n   e   . \r
0000057
```

Here, the end of line marker is a single 'carriage return', `\r`