| | |
|---|---|
| 1. Semester and Year | : 2012 FALL |
| 2. Course Number | : CS-227 |
| 3. Course Title | : Linux & System Software |
| 4. Work Number | : LA-06 |
| 5. Work Name | : Lab 06 |
| 6. Work Version | : Version1 |
| 7. Long Date | : Sunday, 30, September, 2012 |
| 8. Author(s) Name(s) | : Jake Waffle |

## Task 5.1: Plan a Partition Scheme

This task is all about the considerations that go into how one may want his/her system partitioned.

**Understanding x86 Disk Partitioning**
Partitioning on a x86 disk is fairly simple. There are three types of partitions: primary, extended and logical. Our classes book says that primary partitions were originally the only type of partition (Smith, page 192). But because only four primary partitions are allowed on a x86 disk, the extended and logical partition-types were made in order to extend that limit. Extended partitions are able to be substituted for a primary partition and from there logical partitions can allocate a section of that extended partition that's not in use.

Partitioning for Windows is almost unnecessary, unless a dual-boot is in use. Linux is a bit different in its use of partitions. Linux is known to break and need reinstalled (things just go wrong.) So just in case something goes wrong with Linux, its generally good practice to separate the /home and root directory by putting them on different partitions. This way if there is a problem with the root partition, the /home partition will still be intact. Linux also makes use of swap and /boot partitions and can isolate even more directories on their own partition.

**Reviewing Existing Systems**
There are essentially two commands to look up information regarding the sizes of the existing partitions according to our book (Smith, page 192): *df* and *du*. The *df* command when used will display information of space used, space available, mount point and use percentage on all of the existing partitions. And the school computers in room MLH-310 only look to be using the root partition for all of its directories. But with the *du* command along with option -s I can check the space used by the /home directory within the root partition. And the school's /home directory is using 8160 bytes.

**Estimating Disk Space Requirement**
The author of our book basically just addresses system and user data in this section (Smith, page 192). He says, “that the root partitlon, /usr and /boot hold system data, while /home and /var hold user data.“ If there is going to be a lot of user data on your computer, then accordingly allocate the space needed for the user data partitions (/home and /var.)

**My Planned Partition Setup**
I essentially just make partitions for /boot, root, swap, /tmp, /var, and /home when partitioning my Linux systems (I really only care about the /home partition, because I store important things there.)

For a hard-disk with 150gb with two gb of RAM, I would allocate the corresponding amounts for my Linux installation:

/boot = 1 gb
/ = 100 gbs

swap = 4gbs
/tmp = 25gbs
/var = 40 gbs
/home = 80 gbs

## Task 5.2: Use fdisk to Create Partitions

For this task I will be going through the procedure of creating partitions with fdisk.

### Examining the Partition Table

I examined the partition table of my hard disk with command

```
sudo fdisk -l /dev/sda
```

and found that I have four partitions in total including the boot partition. This is for my Windows system entirely though, as I have Ubuntu installed within Windows 7 like as an app almost (it's stored under one of the non-main partitions too.) I just wanted to separate the partitions on my Windows system so that my important files could be stored in their own designated area.

### Deleting Existing Partitions

If a partition is needed to be deleted, it can be done while in fdisk's interactive mode with the command “d”. And if all partitions need to be deleted, the “o” command while in the fdisk's interactive mode can be used.

### Creating New Partitions

There are three different types of partitions that can be created within fdisk's interactive mode: primary, extended and logical partitions. To create any one of these partitions it can be done within fdisk's interactive mode with command “n”.

There are restrictions when creating these partitions. Like for instance, my old Windows XP desktop only allowed four primary partitions when partitioning. And logical partitions can only exist within an extended partition. I do believe extended partitions are also counted as primary partitions (if you have one extended partition, you can only have three primary partitions.)

The swap partition can be made in fdisk's interactive mode too with the command “t”. And after everything is set up to the user's wishes, fdisk needs to be told to write the changes to the disk where the partition table was being edited. This can be done with the “w” command in fdisk's interactive mode.

### Using Advanced Options

Other than what has already been described, our book suggests that there is a lot of other things that can be done with fdisk too (Smith, page 201.) One of the suggested uses is to toggle a partition as bootable with command “a”. And another is to change the ordering of the partitions. But I don't really understand much of these advanced options.

# Task 5.3: Create Filesystems

In this task I'll be going over the different filesystems, the creation of a data filesystem and how to prepare a swap partition for use.

### Deciding What Filesystem to Use

From my research when I was partitioning my Arch Linux system a while back, I found that the Ext4 is a good filesystem to use on your Linux partitions. But it seems that it wasn't around when our book was written. So I'm not sure how to compare it to the ones listed already. But Wikipedia says that Ext4 is a journaling filesystem and those are said by our book to be better for large disks because they are efficient at recovering from file corruption after power failures and system crashes (Smith, page 203.)

I can't really see much of a difference between the filesystems other than the fact that some are journaled systems while others are not. It probably just depends on what you're going after in terms of performance. But for what I needed my other Linux computer for, the Ext4 filesystem worked just fine.

### Creating a Filesystem

Creating a filesystem on a partition is a pretty easy thing to do. It can be done with th *mkfs* tool as explained by the book (Smith, 206.) The command that the book has as an example is

```
mkfs -t ext3 /dev/sda1
```

The -t option denotes that a filesystem of type ext3 will be created on partition /dev/sda1 in this example. So all you need to do is give this tool the intended filesystem and the partition that you intend to place it on.

### Creating Swap Space

Swap space allows the computer to treat disk space as RAM is how the book describes it (Smith, 207.) A simple rule of thumb that I learned a while back was to always create twice as much swap space as there is RAM on the computer.

Creating the swap space can be done by first creating a partition to use and then using the *mkswap* command on it to create the swap space. The command may look like

```
mkswap /dev/sda5
```

where /dev/sda5 is the partition that will become the swap space. One can also use a disk file to create a swap space on with the *dd* command. And then the disk file swap space can be checked to see if it was configured correctly using command *free* to display the amount of swap space and command *swapon* to temporarily activate a given swap space (the swap space can also be turned off with command *swapoff*.) But to permanently activate the swap partition, one must go into /etc/fstab which will be done in task 5.5.

## Task 5.4: Manually Mount and Unmount Filesystems

In this task I will be learning how to mount and unmount filesystems to make them visible on the system or not.

**Mounting a Filesystem**

After learning how to create a filesystem on a partition, we can now mount the partition to a mount point on our computer. But if there isn't a mount point, we can make them with *mkdir.*

```
#mkdir /mountPoint
```

After creating a mount point we are able to mount a partition to it with command *mount.*

```
#mount /dev/sda1 /mountPoint
```

Then we are able to see if the mount was successful with the *df* command. And we are able to focus on a given mount point with the *df* command according to the book (Smith, page 210.)

```
#df /mountPoint
```

**Additional *mount* Options**

A couple of important mount options are emphasized in the book (Smith, page 210-211.) The first is the -t option and it allows on to specify the filesystem of the partition that is being mounted.

```
#mount -t ext3 /dev/sda1 /mountPoint
```

The other is the -a option and it looks up the filesystem for the given partition in /etc/fstab.

```
#mount -a /dev/sda1 /mountPoint
```

**Unmounting a Filesystem**

When unmounting a filesystem with the *umount* command, the book says that either the mount point or the device filename can be given to unmount a filesystem (Smith, page 213.)

```
#umount /mountPoint
or #umount /dev/sda1
```

Options can also be specified when unmounting a filesystem. One worth noting is the -f option that would force an unmount when a filesystem isn't responding.

## Task 5.5: Use /etc/fstab

In this task I will be learning how to set up /etc/fstab to mount certain filesystems at boot time and be able to tell the system about swap space.

**Understanding the /etc/fstab File Format**

Each line within /etc/fstab is separated into six fields according to the book (Smith, page 214.) Those fields are arranged as follows: device, mount point, filesystem type, mount options, backup operation and filesystem check order.

The first four have been discussed in the previous tasks. But the backup operation can be represented with a 0 or 1. The 0 says that the dump utility shouldn't backup the partition and the 1 says that the dump should backup the partition. The filesystem check order column is there so that at boot time, the system knows which filesystems to check with fsck for information regarding their integrity. And it is represented with either a 0 or (1 or 2). A 0 means to not check the filesystem with fsck. And the 1 and 2 both mean to check the filesystem, but the 1 is only for the root partition while all others must use 2.

**Adding a Local Filesystem**

To add a filesystem permanently to a system, one only needs to add the details of that filesystem in an entry line within /etc/fstab in the format that was previously discussed. So adding a line like

```
/dev/sda1       /mountPoint         ext4            defaults        0 2
```

would add the filesystem from device /dev/sda1 with a type of ext4 to the mount point /mountPoint at boot time.

**Adding Swap Space**

The book gives an example for what an entry in /etc/fstab would probably be like for swap space (Smith, page 216.)

```
/dev/sda5       swap            swap    default         0 0
```

The book says that on should normally not deviate from the example entry given. But the device filename is sure to be different in some cases.

**Adding Network Mounts**

I'm not exactly sure what the network mount is able to do after being mounted. But the book says they work similarly to that of the other filesystems (Smith, page 216.) It seems that with a network mount, you mount a server (specified by its server name) to a specific directory so that the server can access that directory. The books example of the entry looks like

```
nfsserver:/optexp       /opt    nfs     ro              0 0
```

And with this example “nfsserver:/optexp” seems to be the server name and /opt seems to be the directory that the server has access to. The filesystem of the server is nfs I think and this is a read-only mount.

**Adding User-Controlled Mounts**

When wishing to allow users to be able to mount specific filesystems on their computers. It is able to be done by adding options to the entries in /etc/fstab that are wanted to be allowed for mounting and unmounting. One option to note is the *users option and it enables ordinary users to mount and unmount the given filesystem. Other options will fit other needs. But this is simple enough.*

# References

-”ext4.” Wikipedia. n.d. Web. 7 Oct. 2012. <http://en.wikipedia.org/wiki/Ext4>

-Roderick W. Smith. linux administrator StreetSmarts. Induanapolis, Indiana: Wiley Publishing, Inc, 2007.