1. Login to your group's Linux guest as root.

2. Create yourself an unprivileged account with a username of your
   choice, for example:
       useradd fred
   If you like, you can choose your favourite shell and set your
   account's full name using the appropriate options to useradd
   (use "man useradd" for details).

From now on, you will have to work in cooperation with the others
in your group since these commands cannot be run by everybody.

3. Your Linux guest has been configured with a readonly link to a
   disk containing the SLES10 distribution. The disk is at device
   number 1010 in your virtual machine. By default, Linux will sense
   all such devices in its virtual machine but will not "bring them
   online" and make them available for use. Bring the 1010 disk online
   by using the command
       chccwdev -e 1010
   The "-e" is for "enable" (what traditional System z O/Ses would call
   "vary online"). The corresponding option to vary offline is "-d"
   (for "disable").

4. When you now issue the command "lsdasd" you will see that Linux has
   brought the device online and the Linux hotplug support has
   assigned and created an appropriate device name for it.

5. The SLES10 distribution files live in an ext2 filesystem on
   partition 1 of that 1010 disk. Create a mountpoint
       mkdir /mnt/sles10
   and mount the filesystem *readonly* with the command
       mount -r /dev/dasdx1 /mnt/sles10
   *replacing the "x" in "dasdx1" with the appropriate letter
   discovered from the lsdasd output.

6. Use the "find" command to look for the rpm packages for Apache:
       find /mnt/sles10 -name '*apache*'
   You will find that SLES10 uses names such as "apache2" for
   Apache 2 packages. Install the Apache software by cd'ing into
   the appropriate directory with the rpm package and installing it
   with
       rpm -ivh apache2......
   filling in the appropriate filename (you can hit the tab key to
   auto-complete partially typed filenames as always with normal
   Linux shells). If you want to install further apache packages
   and modules, then do so (you may need to include other dependent
   rpm package filenames in the "rpm -i" command line). You could
   also use "yast" to provide a "thicker" user interface and handle
   dependency checking and so on for you, but you will need to
   tell yast about the location of your distribution files first.

7. Look at the Apache configuration files underneath /etc (they will
   be in /etc/httpd, /etc/apache2 or similar) to discover where the
   DocumentRoot is. It will probably be /srv/www/htdocs or some such.
   Create a web page called foo.html in there and check that you can
   access it from your browser by using a URL of the form
       http://zeus.moppssc.com/localnet/n/foo.html
   replacing the "n" in "localnet/n" with the final part of your
   guest's LOCALNET IP address. For example, SLES10C1 has IP address
   10.5.0.7 on LOCALNET so you should use "localnet/7/foo.html" at the
   end of your URL. This is because there aren't enough externally
   visible IP address available for everyone so zeus.moppssc.com runs
   a web proxy (Apache's mod proxy) such that access to
   "localnet/n/foo/bar/..." on zeus.moppssc.com map to "/foo/bar/..."
   on LOCALNET 10.5.0.n addresses.

8. If you wish, install your favourite fast and lightweight web
   application environment from the SLES10 distribution (there will not
   be enough memory and diskspace nor lab time to install some of the
   heavier-weight environments such as JBoss) and write and test a
   simple "Hello world" application. (Or just write a CGI program in
   your favourite language instead--you may need to install appropriate

   RPMs for those languages since the base installation on these Linux
   guests is very minimal).

9. If time remains...set up a two-tier or three-tier web application
   environment by cooperating with one or two other groups: one group
   should install and configure a database such as PostgreSQL,
   another should write a little web application (e.g. in PHP or one
   of the Perl or Python web modules) that pulls information back from
   the backend database. If doing a three-tier environment then the
   third group can author the pretty static pages and frontend pages
   which embed the calls to the app server.