Übung Angewandte Bioinformatik
Sommersemester 2010

Linux

This Übung's intention is to introduce you shortly to the ZBH student computer pool. It will cover some essential procedures and programs that are relevant for the following sessions. Whenever you get stuck do not hesitate asking the Übungsleiter.

The ZBH computers are Linux driven. Linux is a free and open source operating system for personal computers. That means, in principle, anybody can download it for free and, if you are a programmer, can manipulate the program sources to your needs. A lot of scientific programs are available for Linux. The ZBH Linux system provides two interfaces for user interaction: a command line interface (text only) and a graphical desktop environment (MS-Windows® like). By the end of this first Übung you should have learnt how to do basic file handling and how to manipulate textfiles in both interfaces.

The first thing you see when you switch on your monitor is the login-screen. It asks you for a username and a password. Linux is a multi-user and multi-tasking system. That means several users can work on several programs simultaneously. To reduce conflicts or misuse, each user has its own home directory. Linux allows you to set read and write permissions for each file and directory individually. By default, only the owner is allowed to write to his/her home directory. Linux is also case sensitive, so Hamburg is different from hamburg.

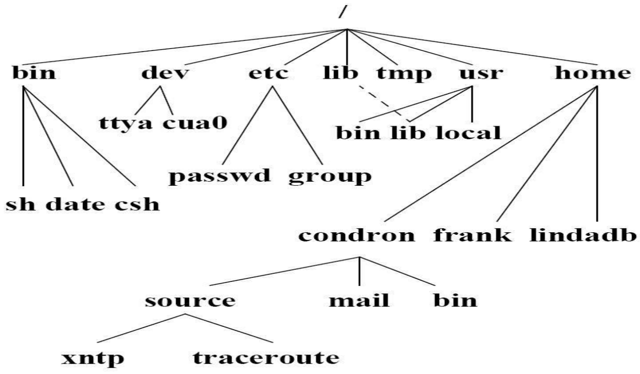
If you have not done so already, you should login now. By default the system loads the K-Desktop-Environment (KDE). Additional to its collection of coordinated programs you can launch any program from within this graphical interface. Most programs can be found in the menu of the taskbar (bottom left of the screen). However, some programs are not listed there. You will learn to use the command line interface in order to launch these programs. This Übung consists of two parts. In the first part you will work with the command line interface, in the second part you will perform exactly the same operations with the graphical interface.

Part I (command line interface):

3) Launch the KDE terminal program "konsole" from the task menu. It provides a command line interface. Whenever you see something like (the command prompt)
   ```
   /home/schenk>
   ```
   the interface waits for a command to be typed in. A command is a program that tells the Linux system to do something
   ```
   command options arguments
   ```
   Options are generally preceded by a hyphen, a `-`, and can be strung together:
   ```
   command -optionoption
   ```
   Arguments tell Linux on what to perform the command, e.g. a file name.
   When you hit the Enter key after typing a command, the system will invoke it.
   If you forget what options and arguments a command needs, take a look at the help manual. It can be launched by
   ```
   man command
   ```
   The navigation is the same as with the `less` command (see question 35).
4) Throughout the rest of this part use the keyboard only. Go to your home directory by typing in the command
   ```
   cd ~
   ```
   Hit Enter to invoke the command. `cd` stands for "change directory" and the argument ~ is an abbreviation of the path to your home directory, e.g. `/home/schenk`. Linux understands absolute and relative paths. We will see later how to use them.
5) Confirm you are in the proper directory by invoking the command
   ```
   pwd
   ```

(`pwd` = print working directory)

6) List all files in your home directory with
   ```
   ls
   ```
7) List all the files in that directory including the hidden files (files beginning with a dot) using the `-a` option of the `ls` command
   ```
   ls -a
   ```
8) List all files including the hidden files in the long format using the `-a` and `-l` options (options may be concatenated)
   ```
   ls -la
   ```
9) The Linux file system is organised as a tree. Note the (sub)directory delimiter is `/`, e.g.
   ```
   /usr/local/
   ```
   You can navigate with the cd command in this tree using absolute and relative paths. An absolute path always starts with `/` (the root directory). There are no `C:` or `D:` drives in Linux, everything can be found under the root directory. A relative path takes into account, where you currently are in the directory tree. For example, suppose you are in your home directory. The absolute path to your desktop would be something like `/home/schenk/Desktop`, whereas the relative path is just `Desktop`.



10) Determine where you presently are in the directory tree with `pwd`
11) Move up one level with
    ```
    cd ..
    ```
12) The `..` is a special relative path pointing one level up no matter where you are in the directory tree.
13) Move to the root directory (`/`) with
    ```
    cd /
    ```
14) Go back to your home directory (`cd ~`).
15) Create a directory `linuxcourse` in your home directory with
    ```
    mkdir linuxcourse
    ```
16) Change into that directory (`cd linuxcourse`)
17) Create a subdirectory *etc* below the `linuxcourse` directory (`linuxcourse/etc`) with
    ```
    mkdir etc
    ```
18) List all the files in the `linuxcourse/etc` subdirectory of your home directory, including the hidden files (`ls -a etc`). The entries you see are symbolic links to the current directory (the `.`) and the superdirectory (the `..`).
19) Remove the `etc` subdirectory in your home directory (`rmdir etc`). ATTENTION: Whenever you remove something in Linux it will be gone immediately and forever. There is no Trash and file recovery in the command line. So be careful with removing.
20) Using the `cd` and the `ls` command, go through some directories on your computer and draw a directory tree similar to the one shown in the figure. Note: You may not have permission to read all directories.
21) In order to determine what permissions you have on the `/etc` directory list the contents of the root directory in long format (`ls -l /`). Find the line that looks like
    ```
    drwxr-xr-x  95 root  root    8192 2009-03-27 12:45 etc
    ```
    The `drwxr-xr-x` tell you what permissions are set. `d` says that `etc` is a directory, the next three characters `rwx` say the permissions set for the owner of the directory, here: Read, Write and eXecutable (for a directory this means the owner is allowed to change into it). The next characters `r-x` and `r-x` are the permissions for the group and the other users, respectively (a `–` means not allowed). To find out which permissions apply to you, you have to know the owner and the group of the directory (here: `root` is the owner and the group is

`root`) and your own username and what groups you are in. Determine your username by `id` and your groups by `groups`. You should find out that you are not `root` and not in the group `root`. That means, you are among the “other users”.

22) What permissions do you have on the `/etc/passwd` file?
23) List all the files and directories in the `/home` directory. Can you imagine what they are? Who has what rights on these files, and what does that mean?
24) Copy the `/etc` directory to the `linuxcourse/etc` subdirectory in your home directory with the copy command. Be aware where in the directory tree you are currently (`pwd`). Change the following command to fit your current situation.

    ```
    cp -r source destination
    ```

    For example, if your are in your home directory. replace `source` with `/etc` and `destination` with `linuxcourse`, if you are in the `linuxcourse` directory use a `.` for `destination` and if you are in the root directory (`/`) you may use the relative path `etc` for `source` and must use the absolute path `~/linuxcourse` for `destination`. If you are anywhere else use the absolute paths for both.
25) Change permission on all files in the `~/linuxcourse/etc` directory and check the result of the commands with `ls -l`. Give yourself read and write permission (`chmod -R u+rw ~/linuxcourse/etc`), people in your group should have read, but no write permission (`chmod -R g+r-w ~/linuxcourse/etc`), and people not in your groups should not have any permission on the files (`chmod -R o-rwx ~/linuxcourse/etc`). The option `-R` stands for recursive and allows you the set also the permissions on all files and subdirectories of `~/linuxcourse/etc` with a single call of the command.
26) Copy the `/etc/group` file to the `linuxcourse` directory in your home directory.
27) Rename the file `group` in your `linuxcourse` directory to `testfile` with

    ```
    mv group testfile
    ```

    Remember to apply the appropriate path like in question 22).
28) Copy the directory `linuxcourse` and all its subdirectories to a directory `work` in your home directory.
29) Remove the directory `linuxcourse` and all its subdirectories (`rm -r linuxcourse`).
30) Rename the directory `work` to `linuxcourse`.
31) Change permissions of the `linuxcourse` directory. Give yourself all permissions (read, write and execute), people in your group should have only the execute permission, and other people should have no permissions at all.
32) Change permissions of all files in the `linuxcourse` directory and all its subdirectories. You should have read and write permission, people in your group only read, other people should be given no permissions at all. You can do that easily with a wildcard. The asterisk (`*`) is wildcard for an arbitrary long character string. Try using the wildcard instead of typing in every filename by hand.
33) Use the `echo` command to print your name on the screen, with a newline before and after your name (`echo “\nyour name\n”`).
34) Display the contents of the file `/etc/group` and `/etc/passwd` using the `cat` command (e.g. `cat /etc/group`).
35) Go to your home directory using `cd` and type in the following command

    ```
    ls –lRF /usr > usrfiles.txt
    ```

    Make sure you type it in exactly like this, watch for the space characters! You will see some “permission denied” messages, ignore these for now. This command will generate a file “`usrfiles.txt`” in your home directory, containing a long listing of files stored on your computer. This may take some time to execute. The > redirects the standard output of the ls command to a file, i.e. Linux creates the file and writes the standard output to this file. If this existed before it will be overwritten. If you do not want this, use >>. This will append the standard output at the end of the file.

36) Afterwards, display the file pagewise using the `less` command. Move down pagewise and linewise, and back up pagewise in the file. You can search in the file for a substring, e.g. blosum, by typing `/blosum` followed by pressing Enter. This will jump to the first occurrence of the substring. To jump to the next occurrence press `n`, to jump back press `N` (Shift-`n`). Then, quit back to the command line by pressing q.

37) Display the first 10 lines of `usrfiles.txt` with

```
head usrfiles.txt
```

38) Display the last 5 lines of `usrfiles.txt` with

```
tail -5 usrfiles.txt
```

39) Move `usrfiles.txt` to the `linuxcourse` folder with

```
mv usrfiles.txt ~/linuxcourse/
```

40) You can avoid the creation of the output file with the pipe symbol |. Let's use the -t option of the ls command to sort its output by time stamp (last modified files come first). Suppose you just want to know which files you last modified. Type in

```
ls -Rlat ~/* | head
```

This redirects the output of the `ls` command to the input of the `head` command.

41) The `grep` command is extremely useful if you just want to display the lines containing, for example, the string “matrix” in the file `usrfiles.txt`. Lets also redirect the long output to `less`.

```
grep matrix usrfiles.txt | less
```

42) Finally, lets do some bioinformatics for a moment. Suppose you have a DNA sequence (e.g. `ATGTTGAGCCTCGCCTAG`) and want to transcribe it to RNA. This can be done with the `tr` command. Go to your linuxcourse directory and create an input file containing a DNA sequence with the `cat` command

```
cat > dna.txt
```

After invoking cat waits for input. Type in a DNA sequence followed by Enter. To leave cat type Ctrl-`D`. Check if a file named `dna.txt` was created and contains your DNA with `ls` and `less`, respectively.

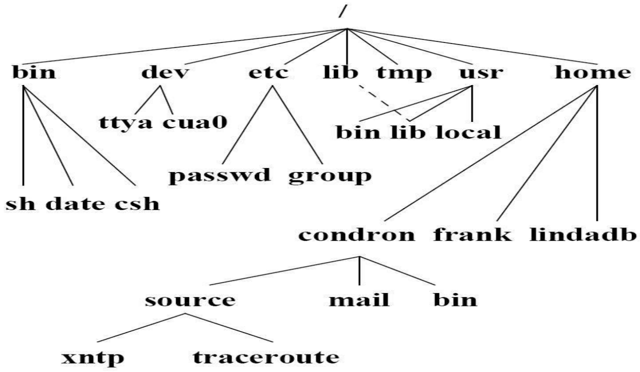
43) In order to transcribe it to RNA you must specify a rule to replace thymine with uracil via two arguments. Use the `tr` command on the file `dna.txt` and save the output in a new file `rna.txt`

```
tr T U < dna.txt > rna.txt
```

Check if the new file exists with `ls` and display its contents with `less`. If you see a RNA sequence you have successfully mastered your first bioinformatics exercise.

Part II (graphical interface)

1) Launch the KDE file and web browser program “konqueror” from the task menu. If you need help using a specific program look for the *Help* menu or press F1.
2) Go to your home directory by clicking on the *home* symbol ⌂ This should directly bring you to your home directory, e.g. `/home/schenk`.
3) Confirm you are in the proper directory by looking at the *Location* toolbar.
4) In the main frame all files in your home directory are listed.
5) List all the files in that directory including the hidden files (files beginning with a dot) using the *View* menu. Tick the entry *Show Hidden Files*.
6) List all files including the hidden files in the long format. Choose in the menu *View->View Mode->Detailed List View*.
7) The Linux file system is organised as a tree. Note the (sub)directory delimiter is `/`, e.g. `/usr/local/`
8) You can navigate this tree with the *Location* bar or by clicking one of these buttons: ← → ↑. A nice feature is the *Tree View Mode* (in the *View* menu). In this mode you can navigate through subdirectories by clicking in the little pluses and minuses.



9) Determine where you presently are in the directory tree with the *Location* bar.
10) Move up one level by clicking ↑.
11) Move to the root directory (`/`) by clicking several times ↑ or by typing `/` followed by pressing Enter in the *Location* bar.
12) Go back to your home directory (⌂) and set the *View Mode* to *Icon View*. Untick *Show Hidden Files* in the *View* menu.
13) Create a directory `linuxcourse` in your home directory by clicking with the right mouse button onto some free space in main frame. The *conext* menu opens. Choose *Create New-> Folder* and type in `linuxcourse` confirm by clicking *OK*.
14) Change into that directory by clicking onto its symbol 📁 (once!).
15) Create a subdirectory `etc` below the `linuxcourse` directory (`linuxcourse/etc`) in the same way.
16) List all the files in the `linuxcourse/etc` subdirectory of your home directory, including the hidden files using the *View* menu. Probably you will not even see the symbolic links to the current directory (the `.`) and the superdirectory (the `..`).
17) Remove the `etc` subdirectory in your home directory. ATTENTION: Whenever you remove something in Linux it will be gone immediately and forever. However, in KDE There is a *Trash* folder, which allows limited file recovery. When right-clicking on the symbol of the `etc` subdirectory you should find an entry named *Move To Trash*. This will move the directory (and all its contents) to the *Trash* folder. It will not delete it. To fully delete it you have to empty your *Trash* folder (can be found on the desktop). This should be done from time to time in order to not mess with our limited disk space.
18) Using the navigation symbols and the *Tree View*, go through some directories on your computer and draw a directory tree similar to the one shown in the figure. Note: You may not have permission to read all directories.
19) In order to determine what permissions you have on the `/etc` directory list the contents of the root directory in *Detailed List View*. Find the column named *Permissions*.
20) The `drwxr-xr-x` tell you what permissions are set. `d` says that `etc` is a directory, the next three characters `rwx` say the permissions set for the owner of the directory, here: Read,

Write and eXecutable (for a directory this means the owner is allowed to change into it). The next characters `r-x` and `r-x` are the permissions for the group and the other users, respectively (a `–` means not allowed). To find out which permissions apply to you, you have to know the owner and the group of the directory (here: `root` is the owner and the group is `root`) and your own username and what groups you are in. See part I.

21) What permissions do you have on the `/etc/passwd` file?
22) List all the files and directories in the `/home` directory. Can you imagine what they are? Who has what rights on these files, and what does that mean?
23) Copy the `/etc` directory to the `linuxcourse/etc` subdirectory in your home directory by right-clicking on its folder symbol and choosing *Copy*. Then navigate to the `linuxcourse/etc` subdirectory and right-click on some empty space. Choose *Paste* from the *context* menu.
24) Change permission on all files in the `~/linuxcourse/etc` directory and check the result by clicking the reload symbol ↻. Give yourself read and write permission, people in your group should have read, but no write permission, and people not in your groups should not have any permission on the files. Right-click on the `etc` folder symbol and choose *Properties*. Click on the tab *Permissions* and change the *Access Permissions* accordingly. Tick the *Apply changes to all aubfolders and their contents* option and confirm with *OK*.
25) Copy the `/etc/group` file to the `linuxcourse` directory in your home directory.
26) Rename the file `group` in your `linuxcourse` directory to `testfile` by right-clicking on its symbol and choosing *Rename*.
27) Copy the directory `linuxcourse` and all its subdirectories to a directory `work` in your home directory (that means copy it and rename the copy).
28) Remove the directory `linuxcourse` and all its subdirectories permanently. Either move it to the trash and empty the trash or hold the Shift-key before right-clicking on the folder symbol and choose *Delete*.
29) Rename the directory `work` to `linuxcourse`.
30) Change permissions of all files in the `linuxcourse` directory and all its subdirectories. You should have read and write permission, people in your group only read, other people should be given no permissions at all. You can do that easily with the recursive option as in question 24).
31) Change permissions of the `linuxcourse` directory. Give yourself all permissions (read, write and execute), people in your group should have only the execute permission, and other people should have no permissions at all.
32) The `echo` command does not work in graphical interface mode.
33) Display the contents of the file `/etc/group` and `/etc/passwd` by navigating to them and choosing *Preview In->Embedded Advanced Text Editor* from the context menu. Go back with ←.
34) Resuse the file `usrfiles.txt` from part I.
35) Display the file using the preview from the context menu. Move down pagewise and linewise, and back up pagewise in the file. You can search in the file for a substring, e.g. blosum, by opening the *Find...* Dialog from the *Edit* menu. Type `blosum` followed by pressing Enter. This will jump to the first occurrence of the substring. To jump to the next occurrence press F3, to jump back press Shift-F3. Then, quit back to the command line by pressing q.
36) There is no `head` or `tail`.
37) Move `usrfiles.txt` to the `linuxcourse` folder using *Cut* and *Paste* from the context menu.
38) Suppose you just want to know which files you last modified. You can sort the file list by clicking on the desired column name in *Detailed List View*.

39) There is no `grep` command.

40) Finally, lets do some bioinformatics for a moment. Suppose you have a DNA sequence (e.g. `ATGTTGAGCCTCGCCTAG`) and want to transcribe it to RNA. This can be done with a text editor. Go to your linuxcourse directory and create an input file containing a DNA sequence by choosing *Create New->Textfile* from the context menu. type in a new filename, e.g. `dna.txt`.

41) Open this file by choosing *Open with->Kedit* from its context menu. Type in a DNA sequence followed by Enter. Save the file with the menu *File->Save* or by pressing Ctrl-`S`.

42) In order to transcribe it to RNA use *Edit->Replace* and type `T` in the *Find* and `U` in the *Replace with* fields. Click on *Find* to highlight the first occurrence of `T`. Click *Replace* to replace this occurrence with `U`. Replace all occurences and close the replace dialog. Save the modified sequence in a new file `rna.txt` via *File->Save as...*

43) Close the editor and check the contents of the new file. If you see a RNA sequence you have successfully mastered your first bioinformatics exercise.