# Brief introduction to Unix

**Lecturer: Prof. dr. Antoine van Kampen (AMC)**

**After reading this chapter you should understand**

* Operating systems
* Unix and Linux
* Basic Unix/Linux commands

## Introduction

Much software used in bioinformatics is developed for the Unix or Linux operating system and does not work on a Windows or Apple computer. Therefore, some basic knowledge about the Unix/Linux operating system is useful. We will also make use of Linux during the exome sequence practicum.

## Computer operating systems

An operating system (OS; Figure 2.1) is a collection of software that manages computer hardware resources and provides common services for computer programs (such as Microsoft Word). The operating system is an essential component of the system software in a computer system. Typical functionality of an OS includes:

* Process management
* Memory management
* File system
* Device drivers (e.g., for your printer)
* Networking
* Security (Process/Memory protection)
* Input and output (etc, keyboard, monitor)

Operating systems can be found on almost any device that contains a computer (from cellular phones and video game consoles to supercomputers and web servers).

Examples of popular modern operating systems include

* Android
* iOS
* Linux
* Microsoft Windows
* Windows Phone



Figure 2.1. The operating system of a computer functions as the interface between a computer application (such as Microsoft Word) and the hardware (e.g., your laptop). (Figure copied from Wikipedia; <http://en.wikipedia.org/wiki/Operating_system>)

## Unix and Linux

[The text below is mainly copied from Wikipedia: <http://en.wikipedia.org/wiki/Unix> and <http://en.wikipedia.org/wiki/Linux>]

**Unix** (officially trademarked as UNIX) is a multitasking, multi-user computer operating system that exists in many variants. From the power user's or programmer's perspective, Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy," meaning the OS provides a set of simple tools that each perform a limited, well-defined function, with a unified filesystem as the main means of communication and a shell scripting and command language to combine the tools to perform complex workflows.

During the late 1970s and 1980s, Unix developed into a **standard operating system for academia**. AT&T tried to commercialize it by licensing the OS to third-party vendors, leading to a variety of both academic (e.g., BSD) and commercial variants of Unix (such as Xenix) and eventually to the "Unix wars" between groups of vendors.

The Open Group, an industry standards consortium, now owns the UNIX trademark and allows its use for certified operating systems compliant with its standard. Other operating systems that emulate Unix to some extent may be called Unix-like. The term Unix is also often used informally to denote any operating system that closely resembles the trademarked system. The most common version of Unix (bearing certification) is Apple's OS X, while **Linux** is the most popular non-certified workalike

*Linux*

Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution. Linux was originally developed as a free operating system for Intel x86-based personal computers. It has since been ported to more computer hardware platforms than any other operating system. It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers: as of June 2013, more than 95% of the world's 500 fastest supercomputers run some variant of Linux. Linux also runs on embedded systems (devices where the operating system is typically built into the firmware and highly tailored to the system) such as mobile phones, tablet computers, network routers, building automation controls, televisions and video game consoles; the Android system in wide use on mobile devices is built on the Linux kernel.

The development of Linux is one of the most prominent examples of free and open source software collaboration: the underlying source code may be used, modified, and distributed—commercially or non-commercially—by anyone under licenses such as the GNU General Public License. Typically, Linux is packaged in a format known as a Linux distribution for desktop and server use. Some popular mainstream Linux distributions include Debian (and its derivatives such as Ubuntu and Linux Mint) and Fedora (and its derivatives such as the commercial Red Hat Enterprise Linux and its open equivalent CentOS).

A distribution oriented toward desktop use will typically include X11 as the windowing system, and an accompanying desktop environment such as GNOME or the KDE Software Compilation.

**In the following some basic constructs of Linux commands are given. This will help you to recognize and understand them during the computer practicum.**

## Some basic Linux (Unix) principles

Although modern Linux distributions allow you to work with software applications in a way similar to Microsoft Windows, much of the scientific (bioinformatics) software is executed from the Linux prompt. For example, the command ‘cd’, which is an abbreviation for ‘change directory’ allows you to go to another directory (which is called a ‘folder’ in Microsoft Windows). For example, the command

> cd /media/Data/Exome

will take me to the directory /media/Data/Exome such that I can inspect and/or use the files in this folder. Note that the ‘>’ represents the Linux prompt in this example.

Once we are in this directory, we can ask for its content by using the command ‘ls’ (list):

> ls

In general, a Linux command takes the form:

> command [-arguments]

where the arguments may be optional. For example, ‘ls’ can take many arguments. One of this is the argument ‘l’:

> ls -l

This will also show the content of the current directory, but provides more details about the files (this is comparable to the ‘detailed’ view in Microsoft windows).

If you need help with a certain Linux command, then you can consult the Linux manual pages. For example, if you want to know which arguments are available for the ‘ls’ command then you can simply type:

> man ls

## Redirection

A commonly occurring situation is that you want to redirect the output of an application or Linux command to a file instead of having the output printed to your screen. This is done by the ‘>’ or ‘>>’ sign. The difference between these two is that ‘>’ will overwrite the file, while ‘>>’ will append the output to the content that is already in the file.

Let us suppose that you want to store the output of ‘ls –a’ in a file. This is very straightforward:

> ls –l > output.txt

Here output.txt is the name of the file. This file can now be printed or further manipulated. For example, we can use the Unix command ‘grep’ to show all lines that contain have the word ‘example’ in their file name:

> grep example output.txt

## Piping

Another powerful mechanism in Unix is ‘piping’. This allows the output from one application to serve as the input of a next application. This is done by using the character ‘|’. Continuing the previous example we could also have done the following:

> ls –l | grep example

## Commonly used Linux commands

man – Display manual page. Most Unix systems have a very comprehensive set of documents. To know how to use a command, use man command name. To search for a command based on a keyword, use man -k keyword.

ls – LiSt the directory content. Filenames that start with a dot are normally hidden for this

command. To see all files including “hidden” files, the option -a must be used. The option -l causes a “long” listing including file ownership, permission etc.

cp – CoPy files. This command requires at least two arguments. The last argument is the

destination and all other arguments are source files. If the destination is a directory, the source file(s) are copied with the same name into the destination directory. If only two arguments are supplied and the destination is a file or doesn't exist yet, the source file is copied as the file with the destination's name.

mv – MoVe files. This is analogous to the cp command, but files are either moved to a new

directory or renamed.

rm – ReMove files. This is used to remove files (not directories). A Unix OS doesn't make a habit of asking if you are sure you want to do something, so use this command with care! If the option -r is used with a directory as an argument, it will recursively remove the complete directory tree. Very dangerous!

pwd – Show the name of the Working Directory.

cd – Change Directory. Change the current directory to the argument or to the users home directory if no argument is given.

mkdir – MaKe DIRectory.

rmdir – ReMove DIRectory. This only works if the directory is empty. To remove a directory

including all its content, use rm -r.

cat – CATenate file(s). Without argument, this will copy stdin to stdout. When filenames are

passed as arguments the contents of all these files are copied successively to stdout.

head – Print the first few lines of either stdin or the file(s) in the arguments to stdout. With the option -n the number of lines to print can be specified.

tail – Analogous to head, but print the last few lines.

wc – Word Count. Count the number of characters, lines and/or words from stdin or files in the arguments.

sort – sort a file or stdin based on its lines.

uniq – remove duplicated lines. With the proper options this can also print only unique or

non-unique lines. uniq expects duplicated lines the input to be consecutive.

tar – tape archive. Create an (uncompressed) archive of a set of files to stdout, a file or a (tape) device.

compress/uncompress – (Un)compress a (single) file or stdin. Because of patents on the LZW algorithm that compress uses, the GNU project developed a patent-free compressor named gzip. This gained much popularity and is a “de facto” standard today.

grep – Search for lines with specified substrings

more – Print the input to the screen, one page at a time