Master ReLAI **Outils informatiques pour la linguistique – M1NS5M2**
Université Bordeaux 3 **2013 – 2014**

TD N^{o}1 - 20/09/2013

---

1 The virtual machine

1. Launch *Oracle VirtualBox* (through the Start menu).
2. Import the virtual machine (`BPL-ReLAI.ova`) in VirtualBox.
3. Examine the virtual machine hardware details (RAM, HDD, etc. . .).
4. Start the virtual machine and, once booted, take a snapshot.
5. Restore the snapshot (optional : break down the virtual machine before restoring).

2 The command-line interface

1. Open a terminal window (console).
2. Check you current folder (`pwd`, `ls [-a]`) and navigate around (`cd [folder]`).
3. Go to the `/root/ReLAI/` folder ; then create a new folder (`mkdir ⟨newfolder⟩`) and then a new file (`touch ⟨newfile⟩`) in it.
4. Rename the file you just created (by moving it : `mv ⟨src⟩ ⟨dest⟩`), then make a copy (`cp ⟨src⟩ ⟨dest⟩`) of it.
5. Remove all the files (`rm ⟨file⟩`) and folders (`rmdir ⟨folder⟩`) you just created.
6. List the contents of the current folder with full details (`ls -la`).
   Note that this command tells you each file's set of permissions (`-rwxr--r--`), owner (`root`), size (in bytes) and last modification date/time.

3 The file system

1. Go to the root of the filesystem (the `/` folder) and list its contents.
2. Ask a search engine (e.g. *google.com, wikipedia.org, etc. . .*) about the linux directory structure. Concentrate on the `/[s]bin`, `/lib`, `/etc`, `/dev`, `/root` and `/opt` folders.
3. Take a look inside the `/opt` folder and display the contents of the configuration file `/etc/profile` using the program `less ⟨file⟩`.
4. Press `q` to quit `less`, then go to the `/root/ReLAI/ana`. . .log-or-something-folder...
5. ...since you don't remember the exact name of the folder that contains the `analog` software, use the terminal's auto-completion feature.
   Just type the first few letters of the expected folder name, then press the `TAB` key. If there is a single appropriate file or folder by this name, the terminal will write its complete name automatically. If there are several, it will display a list of the possible completions if you press the `TAB` key twice. If there are none, it will do nothing. Smart use of this feature can spare you a lot of time under either Linux, MacOS or Windows. (Most modern CLI implement this feature.)

4 Programs under Linux

The `analog` program is not recognized as an installed software by the virtual machine. (Commands like `analog` or `analog-3.3.4` will fail.) It has been developped with the Java language, and must be run using the *Java Runtime Environment* (JRE).

1. Contrary to analog, the JRE is installed and working. Try the command `java -version` and compare with the results of `analog`.
2. In order to find the actual installation of any program, Linux provides the command `which ⟨program⟩` that locates any installed program in the filesystem. Try `which ls`, `which which`, `which analog` and `which java`, and compare the outputs.
3. Go to the folder that contains the `java` program, then list the contents of its parent folder (`ls ..`).

   Most program folders look a bit like this. The `bin/` folder contains the binaries (the programs themselves), `lib/` contains the libraries (dependencies that may be required by the program itself or other programs). Other frequently found folders are `man/` (the user's manual), and `src/` and `include/` (containing the original source code of the program).

   What is the difference between a program or library that is installed and one that is not ? In short, the answer lies in something called environment variables. Without trying to get a full picture, we are going to quickly see what it consists in.
4. The `echo ⟨message⟩` command simply writes a message in the terminal (try it). As an added benefit, it can be used to look at the contents of environment variables, who are special sequences that start with the symbol `$`. Write the contents of the `PATH` environment variable with the command `echo $PATH`.
5. Looking at the output of the last command, try to find an entry (between '`:`' characters) that concerns Java. Is there a similar entry for Analog ?
6. Type in the command `PATH=""`, then press `ENTER`. Try a few commands that you've used before : `echo`, `ls`, `java`, `cd`, etc...
7. Restore the Virtual Machine to a working state. (There are many ways of doing this !)

5 (Bonus) Manually installing software

Most Linux distributions come with a software called a *package manager*, which is capable of automatically downloading, installing and configuring many well-known programs by name ; and installing others from a specific package file. Most package managers are designed to be intuitive and easy to use for newcomers.

Sometimes however, you may encounter software (usually very specific and/or professional-oriented) that do not come in a pre-packaged version and must retrieved, installed, and sometimes compiled by hand. We are going to take a quick look at this process, by compiling and installing by hand an older version of the Python programming language suite.

Before we can do that however, we need to set up the Virtual Machine for it. In order to keep it as light as possible, many standard features of Linux systems are lacking in the VM, including the `gcc` compiler (which transforms C source code into usable software) and the `make` utility used below. This can be easily remedied by loading an additional file containing these tools and loading it at startup. This file can be downloaded from :

`http://distro.ibiblio.org/quirky/precise-5.5/devx_precise_5.5.sfs`

Once placed in the `/` folder, it can be included into the system by going into *Menu* → *Setup* → *Wizard wizard* → *Configure startup of puppy* → *Choose which extra SFS files to load at bootup*, then placing the `devx` file in the right column. This only needs to been done once before the virtual machine is up to speed with other Linux distributions when it comes to compilation. After the next reboot, the machine will be ready for the next step.

First, we need to download the source code of the software, using the `wget ⟨URL⟩` command. In our case, we'll place ourselves in the `/opt` directory and put it there.

```
cd /opt
wget http://www.python.org/ftp/python/2.7.5/Python-2.7.5.tgz
```

Once the download is complete, we get a compressed file (a *tarball*) containing all of the source code. There are two layers in a usual tarball : all of the files are grouped together in a single `.tar` file, which is itself compressed to save space using a software such as `gzip` ; yielding a `.tar.gz` (or sometimes `.tgz`) file. We are going to unpack it entirely with a single command, then step into the newly created folder :

```
tar -xvzf Python-2.7.5.tgz
cd Python-2.7.5
```

The command `tar` is invoked here with four options, abbreviated by letters : x means *eXtract* the contents of the provided tarball ; v is for *Verbose*, displaying a complete list of the files as they are extracted (otherwise, the default Linux policy is to only display messages when there is a problem to report) ; z means that the data has been compressed with *gZip* and should be uncompressed first ; and f tells the program that the tarball it expects is contained in the specified *File*. Most source code releases can be extracted using the exact command above.

The folder we have stepped in contains a bunch of files, among which some have a name in CAPITALS. These files are simple text files, which can be displayed using the `less ⟨FILE⟩` command. Take a quick look at their contents ; you don't need to read all of it (even though you should !). The README file gives the installation procedure, which is almost always as this :

```
./configure
make
make install
```

The `configure` file is a script that looks at the system around it and prepares the recently unpacked source files for compilation accordingly. The `make` software is a very useful tool that, according to the contents of a file named `Makefile`, compiles the specified source code into a binary program, and, with the last command, copies the resulting program into the appropriate system folders for all users to find [1].

Once this is done, the newly installed software is in working order. Just type in the `python-2.7` command, and the Python interpreter should open. Close it using the `CTRL+D` sequence or the `exit()` command.

1. Note that you have to be an administrator, or `root` to perform `make install`. This is always the case with our virtual machine, but on most systems you will need to use something like `sudo make install`