Configuring your Account

October 14, 2014

Below are suggestions which will make your interaction with machines in the lab easier or even safer

**cautious mv or rm or cp** There is a file called `.bashrc` in your home directory (the dot is intentional). Adding the following to it:

```bash
alias rm="rm -i"
alias mv="mv -i"
alias cp="cp -i"
```

will make the commands `mv`, `rm` and `cp` cautious, asking you for a verification whenever the action will make some data disappear, such as when you are about to remove something with `rm` or overwrite something with `mv` or `cp`.

Once added to `.bashrc`, to make the change effective in a shell that is already running, you need to type:

```
source ~/.bashrc
```

Any new shell will get the configuration in `.bashrc`.

**making . part of PATH** The notes mention that whether you need to type `myprog` or `./myprog` is down to whether your so-called PATH variable contains `.` for 'current directory'. Its up to you but if you want to be able to just type the program name then another thing you could added to `.bashrc` is

```bash
export PATH=$PATH:.
```

Same remarks as above apply to making this config change apply.

**position info in emacs** By default when you open some C++ code an emacs mode call `line-number-mode` is activated: if you look at the grey bar at the bottom of the emacs window a number is displayed to show which line you are on. This is one way to make use of the error information from g++ which includes a line number.
You can also execute the emacs command `esc-x goto-line` to move to a particular line.

`g++` gives line-number and position-in-line info for a detected error.

There is a further emacs mode you can activated with `esc-x column-number-mode` which then replaces the line-number display with a pair $l, p$, where $l$ is the line-number, and $p$ is the position (or ‘column’) in the line: emacs counts $p$ from zero, which is 1 different from the `g++` output.

Finally if you really want to go to a particular position in the line (as might happen if it was really long), one way to do it is with repetition trick in emacs where you tell it to repeat something. An example would be `ctrl-u 10 ctrl-f` which will cause it to move forward 10 times. So if you do that at the beginning of a line, it goes to the 10-th character.

**getting ‘accents’ in shell** You may need to do some configuration to have a ‘shell’ do appropriate display of characters encoded using the ISO 8859-1 character codes.

If this is not done, then if you write a program to send for example a char value -42 to the output, you may not see Ø (capital O with an umlaut), although you ought to, because Ø has code 214 according to ISO 8859-1, the base-2 for which represents -42 in 8-bit 2s complement.

- **For a shell run inside emacs**
  you need to make the selections

  Options --> Mule --> Set Language Environment --> European --> Latin-1

  you can do the same thing by typing

  `esc-x set-language-environment`

  and then when prompted

  Latin-1

  This effects only a subsequently started shell: you should kill the current *shell* buffer with `esc-x kill-buffer`

- **for a shell independent of emacs**
  You may have started a shell by clicking as follows

  Applications -> Accessories -> Terminal

  The Terminal program has a menu-bar and you can select ‘Terminal’ in it, then ‘Set Character Encoding’. Select ‘Western ISO-8859-1’ if its there as an option and it is not click ‘Add or Remove’ scroll down through the list of available encodings and select ‘Western ISO-8859-1’ and clicking on the blue arrow should transfer this to the menu of selectable encodings.

  Now if you select ‘Terminal’ from the Terminal program’s menu-bar and then ‘Set Character Encoding’, you can select ‘Western ISO-8859-1’.

  Now running a program which sends a char value of -42 to the output should cause Ø to be displayed.
**further update concerning character encoding** Apart from doing specific exercises concerning writing programs which should output ISO-8889-1 characters, it is probably not a good idea to make the above described configs for ISO-8859-1 always or permanently: there is an unfortunate interaction (of relatively recent vintage) with the way *error* information is given out by `g++` on the lab machines. This uses a different encoding and the above-described ISO-8859-1 configs will make a mess of what should be simple *quote characters.*