Unit: I
Introduction to Vala
Lab session.

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Languages and Computer Science University of Alicante

Content

Development environment

- We are going to work with GNU/Linux. Specifically with the version that EPS installs in the labs.
- We can use the text editor we like: gedit, kate, joe, vim, etc...
- We are going to compile the code in a terminal.

First steps

- Check which tools related with Vala do we have installed.
- Open a terminal, write '`vala`' and without pressing `the space bar` press twice the `TAB` key.
- One of the installed tools must be the Vala compiler: '`valac`'.
- Check the installed version of it: '`valac --version`'

Compile this code

- Create a file 'helloworld.vala' with the next content:

```
/*
* Hello world.
* date: 07/02/2012
*/
class Helloworld : GLib.Object {
  public static int main(string[] args) {
    stdout.printf("Hello World\n");
    return 0;
  }
}
```

- Compile it writing: 'valac helloworld.vala'.
- The result should be an executable file named 'helloworld' that you can run from the terminal by writting './helloworld' and pressing 'enter'. Generate the 'C' code which is actually compiled, using the '-C' option of the Vala compiler.

Debug this code

- Remember that Vala generates an executable file in machine code of the target architecture, not of a virtual machine.
- To debug at a source level code we need to compile with the options: '`--save-temps -g`', that means, something like this: '`valac --save-temps -g helloworld.vala`'.
- To debug the code in a text mode we can use '`gdb`': '`gdb helloworld`'.
- We can also use '`nemiver`': '`nemiver helloworld`'.

Modifications to the previous code

- The previous code gets as parameters of the `main` method the arguments of the program in the '`args`' `array`.
- Modify it so after printing '`Hello World`' prints each of these arguments.
- Modify it one more time so if it does not get any argument as parameter, it asks you the name and after printing '`Hello World`' prints your name.

Exercise (I)

Write an application in Vala that allows the user guessing an integer numeric value in a range between two given numbers, taking into account the following:

1. In the beginning we will ask the user for the minimum and maximum value between which the number to be guessed is.
2. The value to be guessed will be generated in a random way[1].
3. The program will ask the player his/her name and the game will start.

[1] You can check examples in Vala section Mathematics .

Exercise (II)

1. After every trial the program will ask the user if s/he wants to continue trying or prefers to give up. If the user gives up, it will ask him/her if s/he wants the game to finish or to continue, and if the game continues it will ask the user the name of the next player (without changing the number to be guessed) and the new player will continue with the game.
2. After every correct guess of the number, the program will save the name of the player and the number of trials needed to guess the number and it will ask the user if the game continues or stops and in the case it continues it will generate again a random number within the initial range starting again the game and asking the name to the new player.
3. When the game is over it will show the marks obtained by each player, sorted by the number of trials needed (ascending way).

Objectives...

The student knows:

- ☐ Invoke the Vala compiler from the command line.
- ☐ Check the compiler version we are using.
- ☐ Generate executable code with information to be debugged.
- ☐ Use a debug in a textual or graphic mode.
- ☐ Generate 'C' code from Vala code.