LE GRAND MONDE

Ada, un langage pour tout le monde
Par Luke A. Guest

Introduction

Dans cet article je vais aborder le langage de programmation Ada, son historique, ce que l'on peut faire avec et aussi comment vous pouvez l'utiliser sur votre Raspberry Pi. Plusieurs encadrés de couleur contiennent de l'information supplémentaire sur Ada, vous devriez lire ceux-ci afin d'avoir une plus grande compréhension du langage. Alors commençons...

Vous savez probablement à quoi ressemblent les autres langages : la plupart ne sont pas très lisibles et semblent n'être qu'un fouillis de mots (quelquefois même pas des mots) et symboles bizarres. Tous ont différents symboles, mais Ada a été conçu afin que les programmes soient plus faciles à lire par d'autres, même des années après avoir été écrits.

Au contraire de Python ou Ruby, Ada est un langage compilé, semblable au C. Nous devons donc envoyer les programmes Ada (source) vers quelque chose appelé compilateur qui convertit ce code source en langage machine afin qu'il puisse être exécuté directement par l'ordinateur.

Dans cet article, une image système basée sur Debian sera utilisée, donc Squeeze ou Raspbian fera l'affaire. Debian fournit un compilateur Ada, si vous utilisez une autre distribution Linux telle que Fedora, il faudra vérifier sa présence dans le gestionnaire de paquetage. Il s'appelle GNAT, vous savez quoi chercher.

Avant de commencer, je suppose que vous travaillez avec un environnement graphique, comme LXDE (après avoir tapé startx ou démarré directement dans celui-ci, voir page 3 du nº3 du MagPi pour plus d'info), même si dans cet article nous utiliserons un terminal au début.

J'ai vérifié les exemples du texte en me connectant via un terminal distant sur mon Pi.

Début

Avant de commencer à écrire du code Ada et l'exécuter, nous devons installer quelques outils. Nous aurons besoin d'un terminal, d'un compilateur et d'un éditeur de texte, démarrez LXTerminal et tapez la commande suivante :

```
$ sudo apt-get install gnat
```

On vous demandera votre mot de passe, entrez-le et lorsque APT vous demandera si vous voulez continuer, appuyer sur la touche retour afin de laisser APT installer les paquetages.

Après, créez un dossier pour le code source de cet article, nous devrons y accéder pour lancer les commandes directement à partir de lui :

```
$ mkdir -p \
  $HOME/src/baby_steps/lesson1
$ cd $HOME/src/baby_steps/lesson1
```

Créons un nouveau fichier source Ada dans cette fenêtre en tapant ceci dans le terminal :

```
$ nano -w hello.adb
```

Dans nano, tapez le programme 1 (sans les numéro de lignes), tapez **`Ctrl+O`** pour sauvegarder le programme.

Maintenant dans LXTerminal, créez un nouvel onglet avec **`Ctrl-Shift-T`**, ceci ouvrira automatiquement un nouveau terminal dans le répertoire courant utilisé pour ce programme. Nous pouvons maintenant compiler le programme avec la commande suivante :

```
$ gnatmake hello
```

Le programme, gnatmake, est l'interface pour le