Série 5: Fonction (1)

Portée, paramètres, conception

Lien avec le MOOC Initiation à la Programmation (en C++)
Lien avec ICC-Théorie en complément du MOOC

Les éléments du MOOC sur les fonctions sont répartis sur deux semaines. La première partie se concentre sur la conception de fonctions en insistant sur la maîtrise des concepts de portée et de communication entre la fonction et le reste du programme à l'aide de paramètres et de l'instruction return.

La semaine prochaine traitera le sujet de la conception de fonctions récursives. On fournira également quelques outils de mesure du coût calcul effectif pour une exécution donnée d'un exécutable.

Un exercice complémentaire en prévision du projet est fourni en page 2.

Exercices partiels semaine4 du MOOC

- **Document Tutoriel 1ière partie seulement « Calcul de moyenne »**
  - Écriture d'une fonction passant 2 valeurs et retournant leur moyenne

- **Document Exercices semaine 4 du MOOC : 1ière sélection**

  - **Exercice 11 : prototypes (niveau 1, niveau 2 pour questions 4 et 5)**
    - Illustre les bonnes pratiques de vérification des entrées

  - **Exercice 12 : passage de paramètres (niveau 1)**
    - Question : faut-il utiliser le passage par valeur ou par référence ?

  - **Exercice 13 : la fonction cosinus (niveau2)**
    - Décomposition d'un problème en sous-problèmes
    - Chaque sous-problème est traité à l'aide d'une fonction

- **Document Exercices additionnels semaine 4 du MOOC : 1ière sélection**

  - **Exercice 7 : fonctions simples (niveau 1)**
    - Quelle est la bonne pratique à mettre en œuvre concernant les paramètres ?

  - **Exercice 8: Sapin (niveau 2)**
    - Mise en œuvre des grands principes : Abstraction et Ré-utilisation
    - Entrainement pour la décomposition d'un problème

Complément en prévision du Projet

Découverte des outils de manipulation d'image et du format PPM (*in english*)

In this exercise we will learn how to use a third-party-application from the command line. We will do it by using the image manipulation tool called ImageMagick. We will use three of the tools it provides: **display, identify** and **convert.** However; note that ImageMagick has a lot more features than the ones presented here.

- **Displaying an image**
  - Create a new directory **`project`** in your directory **`programmation`**.
  - Open a terminal and go into your **`project`** directory with the **`cd`** command.
  - Download the image file **`epfl.jpg`** in this **`project`** directory.
  - Enter the command **`display epfl.jpg`** to see the image.

    Notice that you have to close the window displaying the image if you want to continue using the command line of the terminal… If you don't want to block the command line while still viewing the image, simply add the character **&** at the end of the command (like in série1), like this: **display epfl.jpg &**

- **Learning image characteristics**
  - To see the format and the characteristics of the image file, you can use the **identify** tool.
  - Run it by entering **`identify epfl.jpg`.** You should observe the following line:

    ```
    epfl.jpg JPEG 942x456 942x456+0+0 8-bit sRGB 58.9KB 0.000u 0:00.000
    ```

    the line above tells you this image is in the **JPEG** format with a *width* of **942** columns and a *height* of **456** lines. It is coded with one byte (**8-bits**) per color in the standard-RGB color space **sRGB**. The file size is **58.9** KB

    One problem we have with this format for our project is that such a file cannot be opened and edited with our usual program editor geany because it is not encoded in alphanumeric characters such as the ASCII code. For this reason we need to convert such a format into the PPM format that can be opened in a standard text editor.

- **Converting image formats**
  - The **`convert`** command is used to make a conversion between different image formats. Let's convert the JPEG image into another format called PGM: **`convert epfl.jpg epfl.pgm`**
  - **`display`** the resulting image. Note that **pgm** is a grayscale image format.
  - Now display the *text content* of the image with the **cat**[1] command: **`cat epfl.pgm`**

    You should observe a lot of strange characters on the terminal. The reason is that the pgm image content was saved in *binary form***.** We are not yet ready for opening it with geany…
  - Let's produce a plain, readable by human, image file where we can read the pixel values in geany. To do this, we should add the **`-compress none`** option of the **`convert`** tool:

    `convert -compress none epfl.jpg epfl.pgm`
  - Display the *text content* of the resulting image one more time with the **cat** command

    Each integer is the grey level intensity of one pixel coded with one byte => values are within [0, 255].
  - Now let's do the same operation to convert "epfl.jpg" to another format called PPM. It is a very similar format to PGM, but it is for color images, not grayscale ones. Run the following command to do the conversion such that we can read the color information for each pixel

    `convert –compress none epfl.jpg epfl.ppm`
    - Display the text content of the resulting image with the cat command.
    - Now three values (Red, Green, Blue) are stored to represent the color of each pixel.
    - convert the epfl.ppm back to the jpg format with the command **`convert epfl.ppm epfl2.jpg`**

      There is no 100% guarantee that it will be exactly the same as the original epfl.jpg image because the jpeg encoding has many parameters and may produce a file that looks the same as the original image but with a slightly different size as can be seen with the **ls** command.
      We'll come back the topic of compression in ICC-theory in the second module of the course.

The project will use the **ppm** format to read an image and a few parameters to transform it. The result will be available in the **ppm** format. You will be able to test your project on your own jpg images with **convert**.

---

[1] cat is standard LINUX command line tool to concatenate and list files. It is also used to display the text file. For its detailed use check its manual page "man cat".