

I Module de gestion d'option, optl

Le développement du module d'option est la partie du projet qui nous a été le plus chronophage. En effet, nous avons dès le début eu pour objectifs de produire un module générique, réutilisable pour de nombreux programmes. A l'instar du module `getopt` qui nous semblait manquer certaines fonctionnalités très importantes tel que la représentation des options par une chaîne de caractères (représentation longue), mais aussi d'autres choses présentes dans les options `linux`.

Dans un premier temps, il nous a fallu effectuer des recherches sur la gestion d'option des programmes `linux`. C'est après ces recherches que nous avons remarqué avec stupeur que les programmes `linux` n'ont pas de norme pour définir leur option. Nous avons donc décidé de nous en créer une en nous inspirant des commandes telles que `cat`, `ls` ou encore `rm` du système `linux`.

Les options peuvent être représentées par deux identificateurs, une version courte et une longue. Les deux ne sont pas obligatoires mais l'un des deux doit au moins exister. Toute option doit permettre d'interrompre le traitement d'option. Toute option peut exiger un argument pour effectuer son traitement. Toutes les options peuvent avoir une description.

1 Définition d'option

1.1 Les options courtes

Une option courte est composée d'un '-' suivi d'un caractère alphanumérique. Pour donner un paramètre à une option courte il suffit de le séparer d'un espace. On peut faire appel à plusieurs options courtes en un seul appel, en suivant le '-' des caractères représentant les options voulues. Cependant, il ne peut y avoir dans cette forme d'appel qu'un seul paramètre demandant un argument et il doit alors être le dernier de la liste.

1.2 Les options longues

Une option longue est préfixée par la chaîne '-'. Pour donner un paramètre à une longue il faut séparer celle-ci de son argument par le caractère '='. <parler de l'autocomplétion des options longues>

1.3 Option help

Nous avons aussi décidé de rendre obligatoire une option, l'option 'help'. Cette option représentée par '-h', '--help', doit afficher une possible description du programme, comment l'utiliser mais aussi la liste de toutes les options suivies de leur possible description. Cette option interrompt le traitement des possibles options suivantes.

2 Implémentation

2.1 Spécification

Pour l'implémentation nous avons décidé de permettre à l'utilisateur de pouvoir modifier certain chose. Notament les préfixe des option courte et longue (On peut notamment envisager des utilisateur originaire windows qui préférerait utiliser le ' ' au '-'), l'indentificateur des deux représentation de l'option 'help'. Nous avons aussi été contraint d'ajouter un spécifieur permettant de garantir que la valeur suivant celui-ci n'est pas une option (lui aussi est modifiable par l'utilisateur du module). Sans ce spécifieur, l'utilisateur ne pourrait pas rentrer la valeur '-help' en prenant cette valeur non pas comme une option mais une valeur à traiter. Ce spécifieur vau par défaut la chaine '-'. De plus, nous avons mis en place un possible traitement sur les éléments qui ne sont pas des options.

2.2 Le code

Pour regrouper toutes les informations nécessaire à la gestion d'une option nous avons donc mis en place un type `optparam` regroupant toutes ces informations. La fonction `opt_init` initialise une instance de ce type. La véritable difficulté à été dans la conception de la fonction de traitement, `opt_parse`. Cette fonction peut être diviser de la façon suivante :

II Bibli

Lien :

<http://www.catb.org/~esr/writings/taoup/html/ch10s05.html>

https://www.gnu.org/software/gawk/manual/html_node/Options.html

https://www.gnu.org/software/libc/manual/html_node/Getopt.html

Pour chaque argument **de** tableau-argument **faire**

Si argument représente NEXT_NOPT **alors**

Traitement de NEXT_NOPT (le spécifieur que la prochaine valeur ne doit pas être considéré comme une option).

Sinon Si argument est une option longue **alors**

Traitement de l'option longue. Pour cela, un appelle a la fonction opt_parse_long vise a trouver l'option dont argument est le seul préfix.

Sinon Si argument est une option courte **alors**

Traitement des possibles options courtes

Sinon

Traitement de ce qui n'est pas une option, à l'aide la fonction hdl_dlt possiblement fournit par l'utilisateur.

Fin Si

Fin Pour chaque

FIGURE 1 – Algorithme principal de la fonction opt_init