

Programmation Avancée

TP n°9 : Les terminaux

Simon Forest

1 avril 2021

Exercice 1 : strtok

1. Écrire une fonction `int est_dans(char c, char *str)` qui renvoie 1 si `c` est un caractère de la C-chaîne `str` et 0 sinon.
2. Écrire une fonction `char* my_strtok(char *str, const char *delim)` qui a le même comportement que la fonction `strtok` vue en cours. Il faudra utiliser une variable `static` pour mémoriser où reprendre la recherche lorsque le premier argument est `NULL`.

Exercice 2 : Terminal

Dans cet exercice, on propose d'écrire un terminal acceptant plusieurs commandes.

1. Écrire un terminal répondant aux commandes de la forme :

```
saluer [nom]
```

```
en affichant
```

```
Bonjour [nom] !
```

2. Étendre le terminal précédent pour qu'il traite des commandes de la forme :

```
mult [n1] [n2]
```

```
en affichant
```

```
Résultat : [n].
```

```
où [n] = [n1] × [n2].
```

3. En utilisant la structure de dictionnaire vue dans le TP n°7, étendre le terminal précédent pour qu'il traite des commandes de la forme :

```
affecter [var] [n]
```

```
et
```

```
afficher [var]
```

qui permettent respectivement d'affecter la valeur `[n]` à la variable `[var]`, et d'afficher la valeur donnée à `[var]` par la dernière affectation.

4. Changer le traitement de `mult` pour que des variables puissent être passées en arguments en plus de nombres.

5. Changer le code du terminal pour permettre de passer en argument des chaînes entre "... " avec possiblement des espaces. Tester la fonctionnalité en exécutant :

```
saluer "Monsieur Marcadet"
```