

TP 2

Loïc LAGADEC
7 Septembre 2021

I Make

1. Mettez en place ce mécanisme (qui sera étendu par la suite) pour les différents exercices du TP.
make est une commande système qui permet de ne compiler que si nécessaire. Pour cela on spécifie dans un fichier *makefile*, illustré par le code suivant, une série de règles.

```
default : exo1 exo2

exo1 : exo1.c TP2.pdf
      gcc -g -Wall exo1.c -o exo1

exo2 : exo2.c TP2.pdf
      gcc -g -Wall exo2.c -o exo2
```

Chaque fichier est associé à une série de dépendances qui sont soit d'autre règles soit des fichiers. Dans le cas de fichiers, si l'un des fichiers listés est postérieur à la cible, alors les commandes associées sont exécutées.

L'appel à *make* se fait en ligne de commande. L'exemple suivant active la règle *exo1*. et l'ensemble des règles.

```
make exo1
```

L'exemple suivant active quant à lui l'ensemble des règles.

```
make
```

Dans l'exemple précédent on tente d'exécuter les commandes associées à *exo1* et *exo2*. *exo1* ne réagit que si le sujet *TP2.pdf* ou le fichier *exo1.c* est postérieur à *exo1*. Dans ce cas, les commandes sont exécutées.

II Tableau et boucles

2. Calculez et affichez la somme des éléments d'un tableau de taille connue
3. Même question en ignorant les éléments négatifs
4. Même question en s'arrêtant à la première valeur négative exclue
5. Recopier un tableau de lettres à l'envers dans une nouvelle variable et afficher les deux textes

III Affichage et appels systèmes

On ajoute une nouvelle librairie pour supporter les appels systèmes

```
#include <stdlib.h>
```

6. Afficher en utilisant des caractères étoile (*) un rectangle de taille *a* par *b* avec *a, b* des entiers
7. Même question pour un triangle rectangle isocèle (on ne conserve que *a*)
8. Ajoutez une temporisation après chaque ligne

```
system("sleep_1");
```

et recompilez et exécutez

9. Ajouter un vidage d'écran avant chaque ligne

```
system("clear");
```

et recompilez et exécutez

IV Pointeurs

IV.1 Manipulation de base

10. Dans le programme précédent, ajouter un pointeur sur a
11. affichez a et pa
12. utilisez pa pour fixer a à 5

IV.2 Manipulation de tableaux

13. Créez un tableau de caractères qui contient

```
[ 'u', 'n', 't', 'p', 'o', 'i', 'n', 't', 'e', 'u', 'r', '\0' ]
```

14. Récupérez son adresse et l'afficher ainsi que la chaîne de caractères
15. Récupérez sa taille en cherchant l'adresse du premier caractère '\0' dans la chaîne
16. Ecrivez '\0' à $addr + 8$; et réaffichez
17. Créez un pointeur sur le 9^{me} élément. Utilisez le pour restaurer le texte initial