

0. Objectifs

L'objectif des deux premiers TMEs est de vous permettre de vérifier que vous maîtrisez les principaux outils de développement utilisés pour programmer efficacement en langage C sous UNIX. Attention : Cette U.E. suppose que vous connaissez le langage C. Si vous ne le connaissez pas, ou si vous avez des lacunes, vous devez vous former par vous même. Les bibliothèques sont faites pour cela, et vous pouvez utiliser les machines des salles de TP en libre service pour faire des exercices.

On utilisera les outils suivants:

- vim ou emacs : éditeur de texte
- indent : outil d'indentation automatique
- gcc : préprocesseur, compilateur, linker
- ar : constructeur d'archives
- make : génération automatique avec gestion des dépendances
- gdb : débogueur

L'application logicielle proposée est une application de manipulation d'images au format Pgm. Les algorithmes réalisent fondamentalement des parcours de tableaux à 2 dimensions.

Commencez par créer un répertoire "tme1", qui contiendra tous les fichiers utilisés dans ce TME.

1. Etape 1 : programme principal

Commencez par créer un sous-répertoire `etape1` dans le répertoire `tme1`, et placez-vous dans ce répertoire. Dans cette première étape, vous allez devoir écrire deux fichiers : un fichier 'main.c' contenant la fonction `main()`, et un fichier 'Makefile' permettant de compiler le programme et de générer un fichier exécutable 'pgmg'.

Le fichier Makefile doit contenir deux règles :

```
pgmg : main.c
```

qui effectue la compilation ET l'édition de liens en une seule passe.

```
clean
```

qui efface les fichiers temporaires et un éventuel 'core', pour ne conserver que les fichiers sources.

Le programme 'main()' va lire le fichier 'input_file_name', en utilisant la fonction `readpgm()`, et recopier chaque octet dans un tampon intermédiaire que vous devez allouer. Il va ensuite recopier le contenu de ce tampon dans un fichier 'output_file_name', en utilisant la fonction `writpgm()`. Le prototype de la fonction `main()` est défini comme suit:

```
int main(int argc, char *argv[])
```

Vous trouverez les prototypes des fonctions de lecture et d'écriture du format pgm dans la page [PgmInputOutput](#).