

TP6 - Tris de Tableaux, 2/2

1 Unix - Ftp

Toujours à la ligne de commande...

1. Récupérez à l'aide de FTP (file transfer protocol) l'archive `tp6.tgz` sur le serveur ftp de polytech Lille, en *anonymous*, dans le répertoire `pub/tp`
 - `ftp ftp.polytech-lille.fr`
 - login : `anonymous`, mot de passe votre email
 - ensuite, `cd pub/tp`
 - `get tp6.tgz`, puis `exit`.
2. Désarchivez l'archive dans votre répertoire de travail pour la programmation structurée. Le répertoire TP6 se crée automatiquement. Il contient entre autres une correction pour le tp5.

2 Visualisation graphique de l'exécution de tris

2.1 Linux : Installation de la librairie graphique

On va utiliser la librairie graphique se trouvant à l'adresse : <http://para.inria.fr/~peskine/enseignement/iup-2002/graphics-doc.html> (lien dispo à partir de la page web du cours).

1. Télécharger la librairie, et trouver un bon endroit pour l'installer (par exemple un dossier `librairies` sur votre *home* (racine)).
2. Extraire l'archive, un répertoire `Libgraphics/` est créé.
3. Aller dans ce répertoire et tapez `make` (compilation de la librairie).
4. Vérifier que tout est bon en tapant `make test`, et en exécutant `./test`.
5. Noter sur un papier le chemin qui mène à `Libgraphics/` : pour l'avoir l'avoir, lancer (par exemple) la commande `pwd` lorsque vous êtes à l'intérieur du répertoire `Libgraphics/`

2.2 Test de la librairie graphique

(Travaillons maintenant dans le répertoire TP6)

Dans le fichier `testgraphic.c`, certaines fonctions de la librairie graphique ont été appelées. Pour pouvoir compiler et exécuter il faut utiliser des commandes qui *lient* la bibliothèque au code C que l'on écrit. Ces commandes sont écrites dans le fichier `Makefile` du répertoire.

1. Ouvrir avec `emacs` ce `Makefile`.
2. Modifier le chemin dans les lignes de configuration en haut : la variable `GRAPHICSLIB` doit contenir le chemin vers le fichier `libgraphics.a` et `GRAPHICSH` doit contenir le chemin vers `graphics.h`¹.

1. la justification de tout cela est dans la feuille annexe de compilation, et sera vue plus tard également

3. La commande `make test` vous permet de compiler un exécutable nommé `testg`. Exécuter.
4. Observer le fichier `testgraphic.c` qui est le fichier compilé lors de la commande `make test`.

2.3 Tri graphique

- Vérifier que le fichier `trisg.c` compile en tapant `make` (la liaison avec la lib graphique est faite dans le Makefile), puis exécuter.
- Dans le fichier `trisg.c`, écrire une fonction `triInsertionGraphique` qui réalise le tri insertion d'un tableau et qui l'illustre sur la fenêtre graphique.

On pourra utiliser une fenêtre graphique de taille 500*500 et la macro suivante :

```
#define RECT(j) (gr_fill_rect(13+10*(j),t[(j)],20+10*(j),10))
```

qui dessine un rectangle proportionnel au nombre contenu à la case $t[j]$ (t , pas un autre nom !)

3 Données expérimentales sous forme graphique : gnuplot

3.1 Prise en main de Gnuplot

1. À partir d'un terminal (répertoire courant TP6), lancer la commande :
`gnuplot gnuplot.gp`
2. Observer les fichiers `gnuplot.gp` et `donnees.dat` (de l'archive), décommentez les lignes de production de fichier dans `gnuplot.gp` et relancer pour imprimer la courbe de points dans un fichier png :

```
gnuplot gnuplot.gp
```

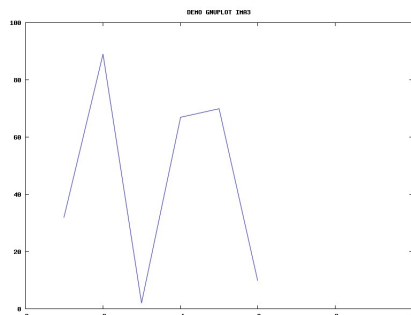
(on pourra utiliser `gqview` ou `evince` pour lire le fichier png généré)

3. Créer un programme C qui demande 6 entiers positifs à l'utilisateur, puis les imprime sur la sortie standard (`printf`) de la façon suivante :

```
1 entier1
2 entier2
...
```

Compiler, tester.

4. Rediriger la sortie standard pour récupérer les informations entrées par l'utilisateur, puis créer un `jpg` avec ces données (cf documentation web de gnuplot) :



3.2 Utilisation avec les données expérimentales

Faire des jolis graphiques avec les temps d'exécution de vos tris (commande `time`).