I. Description :

Une librairie est un sous-ensemble cohérent de fonctions et de programmes "correctement" validés et testés regroupés au sein d'une même archive. Il existe deux formes de librairies : statiques (.a) et dynamiques (.so). Cette page vous donne des exemples avec la compilation de gcc, ça marche aussi avec g++.

II. Librairies statiques :

Pour créer une librairie statique à partir de deux fichiers objets :

```
  $ ar cr libfonc-stat.a fich1.o fich2.o       // -c option de création
$ ranlib libfonc-stat.a                       // crée une sorte d'index (plus vraiment utilisé)
```

Les options liées à *ar* :

```
  $ ar r libfonc-stat.a fich2.o
$ ar a libfonc-stat.a fich3.o
$ ar d libfonc-stat.a fich1.o
$ ar t libfonc-stat.a
```

- -r pour ajouter ou remplacer un fichier objet
- -a pour ajouter un fichier objet
- -d pour détruire un fichier objet
- -t pour afficher l' index des fichiers objets

III. Librairies dynamiques :

Il faut avant tout créer les fichiers objets avec l'option -fPIC ou -fpic, exemple :

```
  $ gcc -fPIC -c fichier.c
```

Ensuite pour créer une librairie dynamique à partir de deux fichiers objets :

```
  $ gcc -shared -o libfonc-dyn.so fich1.o fich2.o
```

Pour finir, pour que lors de l'exécution de l'executable, celui-ci sache où se trouve la librairie, il faut écrire :

```
export LD_LIBRARY_PATH=/chemin-de-la-librairie
```

On peut ensuite exécuter le programme.

IV. La commande nm :

Affiche un index des fichiers objets avec quelques détails :

```
$ nm libfonc-stat.a     // pour une librairie statique
$ nm libfonc-dyn.so     // pour une librairie dynamique
```

L'option *-s* affiche encore plus de détails.

V. Librairies et compilation :

Lors de l'édition des liens, il est nécessaire pour la compilation à l'aide d'une librairie, de dire au compilateur où se trouve cette librairie. Pour ce faire…

```
$ gcc -o  -L -l
```

Exemple lorsque la librairie libvecteur.a que l'on souhaite utiliser se trouve dans le répertoire courant :

```
$ gcc -o test -L. -lvecteur
```

on peut aussi alors écrire directement :

```
$ gcc -o test libvecteur.a   // en statique
```