

Bibliothèques

Sylvain Jubertie

Département Informatique
IUT Orléans

- 1 Introduction
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Plugins

Plan

- 1 Introduction
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Plugins

Bibliothèques

Principe

Une bibliothèque regroupe un ensemble de codes compilés au sein d'un même fichier. Ces codes peuvent être appelés à partir d'autres programmes en les liant avec la bibliothèque correspondante.

Exemples de bibliothèques

libc, OpenGL, Qt, GTK, libjpeg, libpng, libxvid, ...

Intérêts

- compilation + rapide
- factorisation, réutilisation de codes
- organisation : regroupement des fonctions
- maintenance

Bibliothèques

Emplacements sur une distribution Linux

- `/lib`
- `/usr/lib`
- `/usr/local/lib`

3 types

- statique : `.a`
- dynamique : `.so`
- plugins

Plan

- 1 Introduction
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Plugins

Bibliothèques statiques

Principe

Intégrer les codes de la bibliothèque appelés par le programme directement dans celui-ci.

Exemples

`libjpeg.a`, `libmpeg2.a`, `liba52.a`, ...

Bibliothèques statiques

Création

Création de la bibliothèque statique `libtest` contenant des fichiers `.o` à l'aide de la commande `ar` :

```
ar -cr libtest.a file1.o file2.o ...
```

Utilisation

- `libtest` dans le dossier courant :
`gcc -o main main.o libtest.a`
- `libtest` dans un autre dossier :
`gcc -o main main.o -Lpath to libdir -ltest`

Bibliothèques statiques

Avantages

- performance : appel de fonction + rapide que d'appeler une fonction dans une bibliothèque
- portabilité : le programme contient tout le code dont il a besoin et peut être exécuté sur une autre machine sans avoir besoin de la bibliothèque

Inconvénients

- augmentation de la taille de l'exécutable : le code des fonctions de la bibliothèque est dans l'exécutable, donc à réserver à des codes de petite taille
- mise à jour de la bibliothèque statique : recompilation du programme nécessaire pour en profiter

Plan

- 1 Introduction
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques**
- 4 Plugins

Bibliothèques dynamiques

Principe

Au démarrage du programme on va charger les bibliothèques dynamiques utilisées.

Exemples

`libc.so`, `libGL.so`, `libX11.so`, ...

Bibliothèques dynamiques

Création

- 1 Création des objets avec l'option `-fPIC` (Position Independent Code) :
`gcc -fPIC -c file1.c file2.c ...`
- 2 Assemblage des objets dans la bibliothèque dynamique :
`gcc -shared -o libtest.so file1.o file2.o ...`

Utilisation

- `libtest` dans le dossier courant :
`gcc -o main main.c -L. -ltest`
- `libtest` dans un autre dossier :
`gcc -o main main.c -Lpathtolibdir -ltest`

Bibliothèques dynamiques

Avantages

- mise à jour de la bibliothèque : pas besoin de recompiler
- factorisation du code en mémoire : très intéressant dans le cas de bibliothèques de taille importante

Inconvénients

- performance : + lent que le cas statique
- portabilité : il convient de fournir l'exécutable + les bibliothèques statiques (sinon ".dll/.so not found" !)

Bibliothèques dynamiques

error while loading shared libraries!

Causes possibles :

- la bibliothèque n'est pas présente sur le système !
- la bibliothèque n'est pas dans un emplacement standard (/lib, /usr/lib, ...)

Solution 1 : ajouter l'emplacement où se trouve la bibliothèque dans la variable d'environnement

`LD_LIBRARY_PATH`

`export LD_LIBRARY_PATH=pathcontenantleslibs`

Solution 2 : utiliser en tant que root la commande `ldconfig` et le fichier `/etc/ld.so.conf`

Bibliothèques dynamiques

ldd

Pour obtenir les bibliothèques dynamiques utilisées par un programme :

`ldd programme`

Exemple

```
$ ldd /usr/bin/mplayer
linux-vdso.so.1 => (0x00007fff429fe000)
libdirectfb-0.9.so.25 => /usr/lib/libdirectfb-0.9.so.25 (0x00007f373a3d3000)
libXext.so.6 => /usr/lib/libXext.so.6 (0x00007f373a1c2000)
libX11.so.6 => /usr/lib/libX11.so.6 (0x00007f3739ec4000)
libXv.so.1 => /usr/lib/libXv.so.1 (0x00007f37398a0000)
libGL.so.1 => //usr/lib64/opengl/nvidia/lib/libGL.so.1 (0x00007f373a654000)
libdl.so.2 => /lib/libdl.so.2 (0x00007f3738c83000)
libaa.so.1 => /usr/lib/libaa.so.1 (0x00007f3738a65000)
libaudio.so.2 => /usr/lib/libaudio.so.2 (0x00007f37383c2000)
libXt.so.6 => /usr/lib/libXt.so.6 (0x00007f3738160000)
libx264.so.56 => /usr/lib/libx264.so.56 (0x00007f3737635000)
libmp3lame.so.0 => /usr/lib/libmp3lame.so.0 (0x00007f37373c0000)
```

...

Plan

- 1 Introduction
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Plugins**

Plugins

Principe

Cas particulier de bibliothèques dynamiques chargées dynamiquement à l'exécution.

Exemples

`libflashplayer.so`, `mplayerplug-in.so`, ...

Plugins

Création

Comme une bibliothèque dynamique !

Utilisation

Utilisation des fonctions de la `libdl` :

- `void* dlopen(const char* "pathtolib", int mode)`
- `int dlclose(void* handle)`
- `void* dlsym(void* handle, const char* strerror)`
- `char* dlerror()`

Plugins

Exemple

```
#include <stdio.h>
#include <dlfcn.h>
int main() {
    int (*fois2)(int);
    char* err;
    void* handle = dlopen("../libtest.so", RTLD_LAZY);
    if(handle) {
        fois2 = dlsym(handle, "fois2");
        if(!(err = dlerror())) {
            printf("%d", fois2(5));
        }
        dlclose(handle);
    }
}
```

Plugins

Compilation

```
gcc -rdynamic -o main main.o -ldl
```

Plugins

Avantages

- occupation mémoire : chargement uniquement du plugin nécessaire
- mise à jour : ajout/suppression de plugins à l'exécution

Inconvénients

- performance : surcoût du chargement dynamique à l'exécution