



# Édition de liens

Romarc DAVID

LEM2I - Décembre 2011

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- Compilation séparée
- Bibliothèques statiques
- Bibliothèques dynamiques
- Paramétrage de l'édition de liens
- Étude de cas



Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

① La compilation séparée

② Bibliothèques statiques

③ Bibliothèques dynamiques

④ Paramétrage de l'édition de liens  
Statique ou dynamique ?  
Linker scripts

⑤ Étude de cas : bibliothèques mathématiques Intel

⑥ Conclusion



# Compilation séparée I

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- La compilation séparée consiste en la création de plusieurs *fichiers objets* (un par ensemble de fonctions)
- À partir de ces fichiers objets, on peut assembler un exécutable : `gfortran a.o b.o c.o -o monprog`
- Ces processus est l'édition de liens



# Distribution de code objet : un procédé simple I



- Problématique : rassembler les différents fichiers objet.
- Un procédé simple : la bibliothèque statique (archive)
- Assemblage de fichiers objets à intégrer lors de l'édition de liens.
- Édition de liens : résolution des branchements dans le code (appel de fonction = branchement) à la compilation

## Utilisation de ar

- `ar cr archive.a fichier1.o fichier2.o`
- Construction d'un index des fonctions présentes dans l'archive, afin d'accélérer l'édition de liens.
- La mise à jour des composants de l'archive (fichiers objet) est possible individuellement : `ar r archive.a objet.o`

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

① La compilation séparée

② Bibliothèques statiques

③ Bibliothèques dynamiques

④ Paramétrage de l'édition de liens  
Statique ou dynamique ?  
Linker scripts

⑤ Étude de cas : bibliothèques mathématiques Intel

⑥ Conclusion



# Utilisation de la bibliothèque I

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- Utilisation du nom/chemin complet à la compilation
- Indication à l'éditeur de liens : emplacement et nom (⇒ convention sur le nom)

Résultat identique : recopie de l'ensemble du code objet dans l'exécutable.

- Convention de nommage : on parle de bibliothèque (*library*)
- Un nom de fonction présent dans une bibliothèque est appelé un symbole.
- Si l'on nomme l'archive `libmesfonctions.a`
- ⇒ Utilisation abrégée par `-lmesfonctions`
- Indiquer l'emplacement :  
`-L/chemin/vers/la/bibliothèque`



## Utilisation de la bibliothèque II

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

Le compilateur utilise certains chemins de recherche des bibliothèques par défaut. Par exemple, pour le compilateur commercial intel V12, on trouve entre autres

```
/opt/intel/composerxe-2011.1.107/compiler/lib/intel  
/usr/lib/gcc/x86_64-redhat-linux/4.1.2 \  
/usr/lib/gcc/x86_64-redhat-linux/4.1.2/../../../../lib64 \  
/usr/lib/gcc/x86_64-redhat-linux/4.1.2/../../../../lib \  
/lib64 \  
/lib \  
/usr/lib64 \  
/usr/lib \  

```





## Quelques ennuis... I

- Ordre à l'édition de liens
- Dépendances circulaires
- Utiliser le groupement de bibliothèques `-Wl,-start-group`  
`a.a b.a -Wl,-end-group`

# Caractéristiques des bibliothèques statiques I

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- Elles conduisent à un exécutable auto-suffisant (appelé exécutable statique)
- Ainsi, l'exécutable est plutôt indépendant de l'état du système (et de l'environnement du processus)
- Inconvénient : toute modification dans la bibliothèque (correction de bugs ...)  $\Rightarrow$  exige une recompilation de code final de l'utilisateur !
- Si plusieurs codes utilisent les mêmes bibliothèques, le code de ces bibliothèques est présent plusieurs fois en mémoire.





# Plan I

- 1 La compilation séparée
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Paramétrage de l'édition de liens  
Statique ou dynamique ?  
Linker scripts
- 5 Étude de cas : bibliothèques mathématiques Intel
- 6 Conclusion

## Distribution de code objet : un procédé élégant I



- Création de bibliothèques dynamiques (option `-shared` sous Linux et `-dylib` sous Mac Os X)
- Compilation du code avec références à ces bibliothèques (options `-L` et `-l` , identiques à la compilation statique)
- `-L` indique l'emplacement des bibliothèques lors de la compilation du programme
- `-ltruc`  $\Rightarrow$  Fichier `libtruc.so`. Il s'agit d'un raccourci syntaxique
- on peut aussi indiquer directement `/usr/lib/libtruc.so.1` par exemple
- À l'exécution, calcul des branchements
- Si plusieurs programmes utilisent les mêmes bibliothèques, ils se partagent le code de ces bibliothèques.

# Exécution d'un programme dépendant de bibliothèques dynamiques I

- Visualisation des dépendances : `ldd`
- Variables d'environnement `LD_LIBRARY_PATH`  
(utilisateur)
- Chemins de recherche standard du système :  
`/etc/ld.so.conf`
- `/usr/lib` puis `/lib`





# Exécution d'un programme dépendant de bibliothèques dynamiques II

ld.so parcourt successivement tous les emplacements indiqués à la recherche des bibliothèques dont dépend le programme (cette information provient de la compilation du programme). Si une bibliothèque vient à manquer, un message du type

```
./adyn : error while loading shared libraries :  
liba.so : cannot open shared object file : No such  
file or directory
```

 est affiché et l'exécution du programme s'arrête.

# Environnement d'exécution I

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

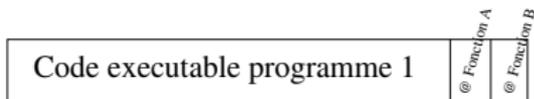
Problème : comment fixer à l'exécution le chemin de recherche des bibliothèques ?

- Pour le chargeur dynamique (`LD_LIBRARY_PATH`), dans l'environnement utilisateur
- Embarqué dans l'exécutable :
  - `-rpath` à l'édition de liens
  - Plus directement `-Wl,-rpath=...` à la compilation
  - ou variable `LD_RUN_PATH`



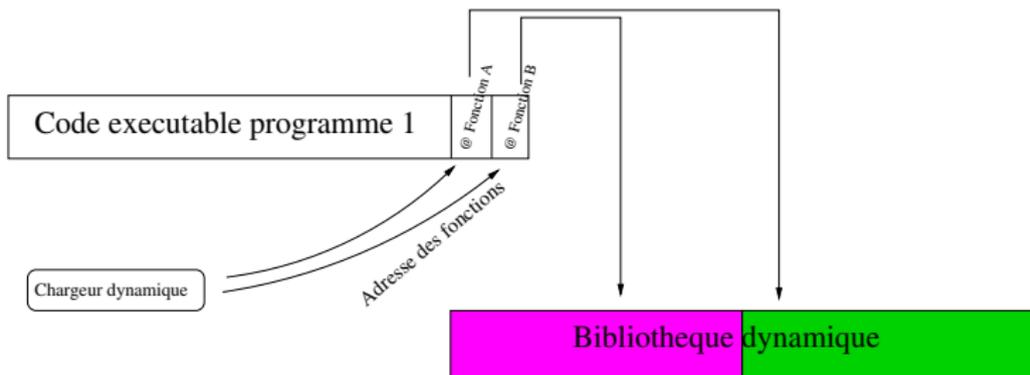


# The big picture





# The big picture



# The big picture

Plan

La compilation séparée

Bibliothèques statiques

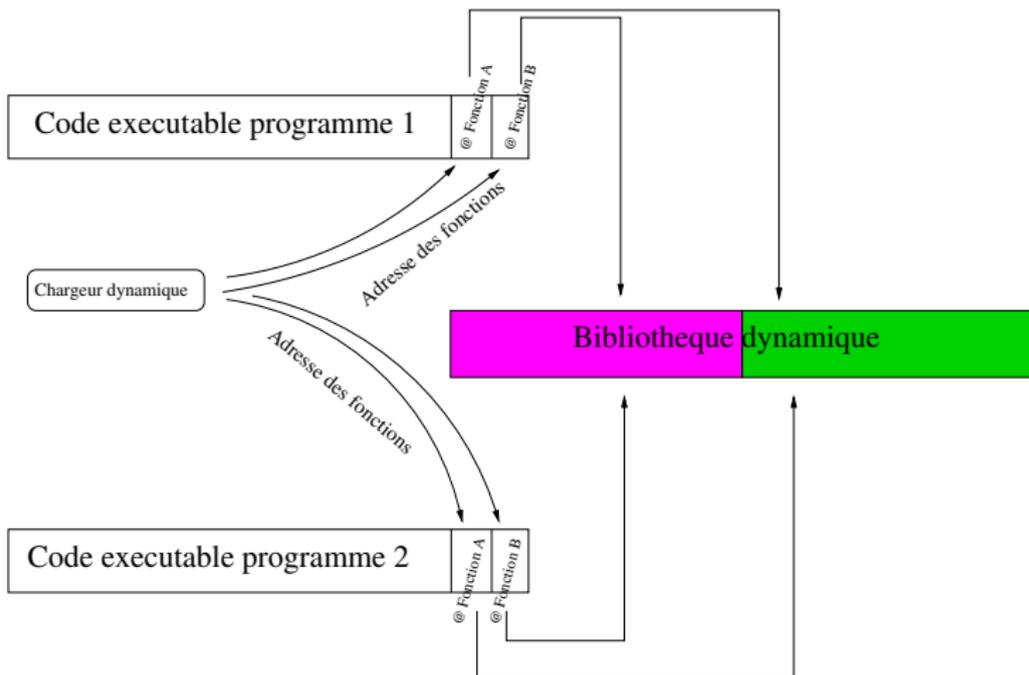
Bibliothèques dynamiques

Paramétrage de l'édition de liens

Statique ou dynamique ?  
Linker scripts

Étude de cas : bibliothèques mathématiques  
Intel

Conclusion



Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- 1 La compilation séparée
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Paramétrage de l'édition de liens**  
Statique ou dynamique ?  
Linker scripts
- 5 Étude de cas : bibliothèques mathématiques Intel
- 6 Conclusion



# Statique ou dynamique ? I

- Par défaut, choix de la bibliothèque dynamique
- Option `-Wl,-Bstatic` : les bibliothèques indiquées après cette option seront liées statiquement.



# Fonctionnalités avancées I

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- Mise à jour du système facilitée
- Bibliothèques d'interposition : permettent d'intercepter les appels (debug, profiling). Exemple : vampirtrace





# Scripts du linker I

- Tout fichier n'étant pas une archive ou une bibliothèque dynamique sera interprété comme un script
- Script : ensemble de commandes destinées à l'éditeur de liens
- Commandes `INCLUDE` ou `GROUP`

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- 1 La compilation séparée
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Paramétrage de l'édition de liens  
Statique ou dynamique ?  
Linker scripts
- 5 Étude de cas : bibliothèques mathématiques Intel
- 6 Conclusion





# Aperçu de la bibliothèque I

Les dépendances entre les différentes composantes peuvent se représenter comme ceci :

# Aperçu de la bibliothèque II

## Intel MKL 10.0

Plan

La compilation séparée

Bibliothèques statiques

Bibliothèques dynamiques

Paramétrage de l'édition de liens

Statique ou dynamique ?  
Linker scripts

Étude de cas : bibliothèques mathématiques Intel

Conclusion



`[-cluster components]`

`[-lmkl_solver] [-lmkl_solver_ilp64] [-lmkl_solver_ilp64]`

`[-lmkl_lapack95]`

`[-lmkl_blas95]`

`[-lmkl_lapack]`

`-lmkl_ia32`

`-lmkl_em64t`

`-lmkl_ipf`

`-lmkl_intel_ilp64`

`-lmkl_intel_lp64`

`-lmkl_intel_sp2dp`

`-lmkl_gnu_ilp64`

`-lmkl_gnu_lp64`

`-lmkl_intel_thread`

`-lmkl_gnu_thread`

`-lmkl_sequential`

`-lmkl_core`

`[-lguides]`

`[-liomp5]`

`[-lpthread]`

`[-lm]`

# Deux exemples de compilation I

Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- Version parallèle (threads) de la bibliothèque avec le compilateur intel : `-L/chemin-mkl -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core`
- Version séquentielle `-L/chemin-mkl -lmkl_intel_lp64 -lmkl_sequential -lmkl_core`



Plan

La compilation  
séparée

Bibliothèques  
statiques

Bibliothèques  
dynamiques

Paramétrage  
de l'édition de  
liens

Statique ou  
dynamique ?  
Linker scripts

Étude de cas :  
bibliothèques  
mathématiques  
Intel

Conclusion

- 1 La compilation séparée
- 2 Bibliothèques statiques
- 3 Bibliothèques dynamiques
- 4 Paramétrage de l'édition de liens  
Statique ou dynamique ?  
Linker scripts
- 5 Étude de cas : bibliothèques mathématiques Intel
- 6 Conclusion





## En résumé...

- Construction de bibliothèques : tâche facile
- Codes parallèles : préférer les bibliothèques dynamiques
- Nécessite environnement utilisateur bien huilé
- Porte ouverte à l'interfaçage avec d'autres langages, comme python par exemple