

TD 2
22/3/00

Objectifs : Comprendre les différentes étapes de la traduction d'un programme.

Loguez-vous sous unix.

Créez un répertoire TD2. Copiez-y les fichiers qui sont dans /home/users/TPINFO/sys1.

Les fichiers mymain.c et util.c constituent un programme. Le fichier makefile vous permettra de voir toutes les étapes de la traduction en exécutable en tapant

> *make nom_du_fichier_à_générer*

(sous xEmacs, utilisez pour cela la commande de menu tools/compile). L'exécutable à générer s'appelle mymain.

1. Lire le code en C. Que fait ce programme ?

Quelles sont dans chacun des fichiers :
les références externes ?
les variables globales ?
les variables locales ?

2. Générez les fichiers mymain.i et util.i. Notez les commandes de compilation produites par make. :

Dans les fichiers .i, ou sont les directives (#include, #define) :

Quel est l'effet de la directive #include ? Qu'est devenue la ligne 21 de main.c ?

3. Générez util.s. Notez les commandes de compilation produites par make.:
Peut-on retrouver dans ce fichier l'endroit où il y a une référence externe ?
Peut-on retrouver les noms des variables locales ?

4. Générez mymain.s.
Mêmes questions qu'en 3.

Peut-on retrouver la variable globale ?

5. Modifiez mymain.c en initialisant i (écrire `int i = 0` ; à la ligne 16). Générez à nouveau mymain.s. Qu'y a-t-il de changé ?

6. Essayez les commandes `cc util.c` et `cc mymain.c` (par `tools/compile` ou dans une fenêtre de commandes). Expliquez le résultat :

7. Générez mymain (`make -k mymain`). Quelle commande de compilation a été générée ?

Comparez la taille du fichier mymain à celles de mymain.o et util.o :
D'où vient la différence ?

8. Écrivez un programme très court qui ne fasse aucune référence externe, par exemple :

```
void main() {  
    int i=0;  
    while(i<5) i++ ; }
```

Compilez-le. Comparez la taille du fichier objet et la taille de l'exécutable :
Expliquez la différence :

9. Renommez le fichier makefile et créez un nouveau fichier makefile dans lequel vous ne recopiez que les 2 dernières lignes de l'ancien. Effacez tous les fichiers .i, .s et .o.
Maintenant, exécutez `make -k mymain`. Notez les fichiers qui restent dans votre catalogue à la fin :

10. Copiez les fichiers mymain.c et util.c sur une disquette. Redémarrez sous Windows. Copiez les fichiers dans un catalogue td2 et ouvrez-les sous Borland.
Créez un projet. Appelez-le mymain.ide (les 3 autres cases de choix doivent contenir : **application**, **win32**, **console**). Cliquez advanced et choisissez c-node, puis dé-cochez .rc et .def . Validez le tout.

Dans la fenêtre de projet, cliquez avec le bouton droit sur mymain.exe. Choisissez add node, et sélectionnez util.c. Dans le même menu, choisissez Edit local options, et positionnez les directories Source, Intermediate, Final sur td2 (le nom complet).

Toujours dans la fenêtre de projet, cliquez avec le bouton droit sur util.c. Choisissez Special/CCompile. Faites de même avec mymain.c. Regardez les fichiers .asm obtenus. Sont-ils les mêmes que ceux obtenus sous unix ? expliquez.

Cliquez avec le bouton droit sur mymain.exe, choisissez makenode.
Notez la taille des fichiers obtenus et comparez-la à celle des fichiers obtenus sous unix.