

Projet

1 Introduction

L'objet de ce projet est d'écrire la commande `maque`, similaire à la commande `make`, permettant de lancer plusieurs processus exécutant les commandes contenues dans un fichier `Maquefile`.

Vous rendrez un rapport de quelques pages (8 maximum) présentant vos solutions aux problèmes posés.

2 Présentation de l'analyseur de fichier `Maquefile`

Un analyseur de fichier `Maquefile` est donné sur la plateforme Moodle. Il construit un arbre des dépendances entre les différentes cibles.

Les deux fichiers source `maquefilereader.c` et `maquefilereader.h` fournis vous proposent les principales fonctions de lecture de `Maquefile` et de construction du graphe de dépendances. Ils sont documentés (Doxygen), voir le source pour plus d'information.

Pour vous aider à vous familiariser avec cette bibliothèque, le fichier `tmfr.c` contient un exemple de lecture d'un fichier `Maquefile` et d'affichage du graphe de dépendances.

Vous pouvez modifier la bibliothèque `maquefilereader` si nécessaire.

3 Réalisation de `maque`

Écrivez un programme en C, utilisant la bibliothèque `maquefilereader` fournie, qui lit un fichier `Maquefile`, puis exécute les commandes nécessaires à la construction de la cible principale¹ et de toutes les cibles intermédiaires nécessaires. À ce stade, ne tenez pas compte de l'option « `-j` ».

Les commandes seront exécutées à l'aide de la primitive système `exec` (ou une de ses variantes) de manière à exécuter « `/bin/sh -c <commande>` ». La construction des cibles s'arrêtera dès qu'une des commandes échoue. Les dates des fichiers de dépendance et de la cible (s'ils existent) seront vérifiées pour ne reconstruire que ce qui est nécessaire, comme le fait la commande `make` standard. D'autre part, si une cible est reconstruite, toutes les cibles qui en dépendent sont reconstruites.

4 Parallélisation

Implémentez l'option « `-j` » permettant de spécifier le nombre maximal de processus que `maque` peut créer simultanément². Les cibles indépendantes seront reconstruites par des processus différents, mais toujours dans l'ordre pour les commandes associées à chaque cible bien évidemment.

Pour vous faciliter la tâche, vous pouvez créer une liste des cibles à construire dans l'ordre de construction, en faisant un parcours *en largeur* du graphe de dépendances.

Dans l'exemple suivant, les trois cibles « `a` », « `b` » et « `c` » peuvent être construites simultanément :

```
all: a b c
    echo "done"

a: a.c
    gcc -c a.c
    gcc -o a a.o
```

1. La cible principale est la première cible trouvée dans le fichier si elle n'est pas spécifiée sur la ligne de commande.

2. On ne tiendra pas compte des processus supplémentaires qu'une ligne de commande peut éventuellement générer.

```

b: b.o
   gcc -o b b.c
b.o: b.c
   gcc -c b.c
c: c.c
   gcc -o c c.c

```

Attention aux dépendances inter-cibles et aux dépendances dans les commandes exécutées :

- la suite des commandes permettant de construire la cible « a » doivent être exécutées dans l'ordre (par exemple, « gcc -o ... » n'est exécuté qu'après que « gcc -c ... » se soit terminé) ;
- la cible « b » ne peut être construite qu'après que la cible « b.o » ait terminé sa construction. Par contre les cibles « a » et « c » sont indépendantes ;
- la commande « echo "done" » ne doit être exécutée qu'après que les trois dépendances de « all » se soient bien terminées.

5 Temps d'exécution

Ajoutez dans votre programme la prise en charge d'une nouvelle option « -T ». Cette option permet d'activer la mesure et l'affichage des temps d'exécution (real/user/sys) de tous les processus lancés pour construire chaque cible : la commande `make -T` devra afficher le temps (real/user/sys) pour chaque cible construite en incluant les temps de construction de ses dépendances.

6 Mesures de performances

Vous ferez des mesures de performances en compilant le logiciel `xfig` fourni sur moodle. Attention, le Makefile donné suppose que vous compilez ce projet sur la machine turing.

Représentez sur un graphe les temps (réel et CPU) obtenus pour différentes valeurs significatives de l'option "-j". À partir de quelle valeur l'augmentation du nombre de processus ne procure plus de gain significatif ?

Vous prendrez soin de fiabiliser vos données en prenant plusieurs mesures et en écartant les données aberrantes. Vous documenterez votre méthodologie de mesure dans votre rapport.