

Durée : 2 heures

Ni les documents, ni les calculettes ne sont autorisés.

Les téléphones portables doivent être éteints.

1 Options de compilation

Soit *programme.c* un code source C du répertoire courant. Donner les options de compilation nécessaires pour générer le code exécutable *programme* quand *programme.c* contient les instructions suivantes et fait effectivement appel aux fonctions de la librairie mathématique et de la librairie X11.

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <X11/Xlib.h>
#include <X11/keysym.h>
#include <Xutil.h>
```

Les fichiers *stdio.h*, *string.h* et *math.h* sont dans le répertoire `/usr/include`. Les fichiers à inclure pour X11 se trouvent dans l'un ou l'autre des deux répertoires `/user/openwin/share/include` ou `/user/openwin/share/include/X11`. Les fonctions de la librairie mathématique sont dans l'archive `/usr/lib/libm.a`; celles de la librairie X11 se trouvent dans `/usr/openwin/lib/libX11.a`.

On désire de plus :

- inclure le fichier `/projet/include/declaration.h` qui a été omis dans le code source.
- définir la macro `DEBUG` avec une valeur égale à 3.
- pouvoir faire appel à un déboggeur symbolique tel que `gdb` pour contrôler l'exécution du programme.

2 Makefile

Soit le fichier *makefile* suivant :

```
OBJETS=global.o recalage.o geometrie.o interface.o
COMP=gcc
CFLAGS=-c
```

```

corr2: $(OBJETS)
        $(COMP) -o cor2 $(OBJETS)

$OBJETS : defs.h

.c.o:
        $(COMP) $(CFLAGS) $< -o $@

clear:
        rm $(OBJETS)

```

1. Donner les instructions exécutées lors de l'appel à *make* (sans arguments) quand seuls les fichiers *global.c*, *recalage.c*, *geometrie.c*, *interface.c* et *interface.o* existent et sont à jour dans le répertoire courant.
2. On suppose que les cibles *corr2* et *\$(OBJETS)* sont temporairement permutées dans le fichier *makefile*. Donner les instructions exécutées lors de l'appel à *make* (sans arguments) quand les conditions d'exécution sont identiques à celles de la question précédente.
3. Ajouter dans le fichier *makefile* l'appel à la librairie mathématique dont on vient de se rendre compte qu'il a été omis.
4. Modifier le fichier *makefile* pour que la génération du fichier *interface.o* dépende également d'un fichier *inter.h* du répertoire courant.
5. Modifier le fichier *makefile* pour définir une macro *SRC* contenant tous les fichiers source C du répertoire courant et calculer la valeur de *OBJETS* en fonction de *SRC*.

3 Réseaux de pétri

Utiliser le formalisme des réseaux de Pétri pour modéliser un système comportant une unité de production et une unité de consommation. Ces 2 unités sont reliées entre elles par un entrepôt (ou magasin) à accès exclusif et ne peuvent fonctionner simultanément (le système est soit en phase de production, soit en phase de consommation de façon non forcément cyclique).

Le fonctionnement de l'unité de production est tel que le système passe de l'état AP (attente de production) à l'état P (production) puis D (les pièces produites sont placées dans un dépôt local). Les pièces ne sont transférées que dans un deuxième temps du dépôt local à l'entrepôt commun lorsque le système retourne à l'état AP.

Le fonctionnement de l'unité de consommation est tel que le système passe de façon cyclique de l'état AC (attente de consommation) à l'état R (retrait des pièces de l'entrepôt) puis C (consommation des pièces en local).

On suppose que :

- la capacité de l'entrepôt est N,
- les 2 unités ne peuvent accéder simultanément à l'entrepôt,
- l'unité de production ne peut livrer une pièce à l'entrepôt que si celui-ci n'est pas plein,
- l'unité de consommation ne peut prendre une pièce dans l'entrepôt que si celui-ci n'est pas vide.

Initialiser votre réseau de telle sorte qu'il y ait un producteur, deux consommateurs et que l'entrepôt soit initialement vide, de capacité 4.