

Systèmes Embarqués 1 & 2: Travail écrit no 4.

Nom :

Prénom :

Classe : T/2

Date : 12.06.2014

Problème n° 1 (programmation orienté-objet)

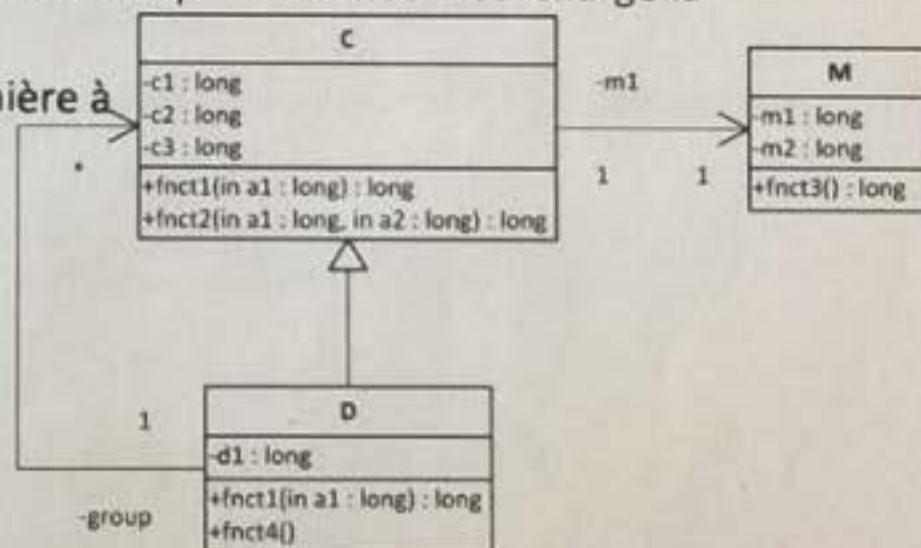
1. Décrivez succinctement le principe d'orienté-objet en langage C.

2. Décrivez succinctement le mécanisme permettant à une méthode d'une classe dérivée d'obtenir la référence sur son objet (l'objet dérivé) à partir de la référence sur l'objet de la classe de base.

3. Pour le diagramme de classes ci-contre :

a. Déclarez les classes C, D et M en langage C orienté-objet. Remarque : la classe D surcharge la fonction «fnc1» de la classe C.

b. Implémentez la fonction «fnc1» de la classe D de manière à ce qu'elle retourne le produit de « a1 * d1 * c1 »



Systèmes Embarqués 1 & 2: Travail écrit no 4.

Problème n° 2 (Toolchain)

1. Expliquez succinctement le principe de fonctionnement d'un Makefile
2. Décrivez la fonction du mot clef « .PHONY: »
3. Définissez une règle permettant de générer les fichiers objets « .o » à partir de leur source « .c »
4. Décrivez la fonction de la règle `include $(OBJDIR :.o=.d)`
5. Indiquez la signification de la commande « `$(MAKE) -C <dirname> <targetname>` » placée dans un Makefile:
6. Indiquez la signification de la commande shell « `$> make HOST=apf27 clean all` »:
7. Indiquez en une phrase la méthode pour déboguer une application fonctionnant sur une cible à partir d'une machine hôte.
8. Indiquez la fonction des utilitaires ci-dessous:
 - a. gcc :
 - b. as :
 - c. ld :
 - d. ar :
 - e. strip :

Systèmes Embarqués 1 & 2: Travail écrit no 4.

Problème n° 3 (Vérification)

1. Citez 2 mécanismes permettant de valider la bonne réalisation et le bon fonctionnement d'applications logicielles durant les différentes phases de leur développement
2. Citez une méthode permettant d'identifier le goulet d'étranglement dans une application logicielle. Citez l'utilitaire de la chaîne d'outils GNU permettant de mettre en œuvre cette méthode.
3. Citez une méthode permettant de garantir qu'un composant logiciel a été correctement testé. Citez l'utilitaire de la chaîne d'outils GNU permettant de mettre en œuvre cette méthode.
4. Décrivez succinctement le principe des tests unitaires.
5. Implémentez un test unitaire permettant de valider/vérifier le bon fonctionnement de la fonction « `double sqrt (double x)` » ci-dessous (2 tests positifs et 2 tests négatifs).


```
/*
 * The sqrt() function computes the square root of x.
 * The following error conditions are defined for this function:
 *   - if x is NAN: NAN is returned and errno is set to EDOM
 *   - if x is negative: NAN is returned and errno is set to EDOM
 */
double sqrt (double x);
```

Systemes Embarqués 1 & 2: Travail écrit no 4.

Problème n° 4 (Documentation & DMA)

1. Citez 4 éléments importants devant faire partie de la documentation d'un logiciel pour le développeur
2. Citez et décrivez succinctement un logiciel permettant de documenter du code source en langage C
3. Décrivez succinctement l'utilité d'un SCM (Source Code Management Tool) tel que GIT ou SVN
4. Décrivez succinctement la fonction d'un DMA
5. Citez les 4 phases principales d'un transfert DMA entre une mémoire et un périphérique
6. Citez 2 domaines d'application des DMA

Systemes Embarqués 1 & 2: Travail écrit no 4.

Problème n° 5 (Mémoire cache et MMU)

1. Décrivez succinctement la fonction de la mémoire cache
2. Citez et décrivez succinctement les deux principes qui sont à l'origine des mémoires caches
3. Décrivez succinctement le principe des mémoires caches complètement associatives
4. Citez 2 algorithmes utilisés pour le remplacement des lignes dans la mémoire cache
5. Décrivez succinctement le fonctionnement de la MMU et citez ses 3 fonctions principales
6. Décrivez succinctement la traduction d'une adresse virtuelle en adresse physique
7. Citez le mécanisme implémenté par les MMU pour accélérer la translation d'adresses virtuelles en adresses physiques