

Microprocesseurs 1 & 2: Travail écrit no 4.

Classe : T/2

Nom :

Prénom :

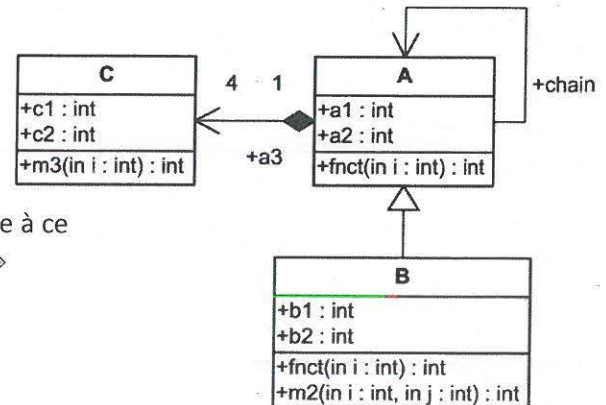
Date : 11.06.2012

Problème n° 1 (programmation orienté-objet)

1. Pour le diagramme de classes ci-contre :

a. Déclarez les classes A, B et C en langage C orienté-objet.
Remarque : la classe B surcharge la fonction «fnc» de la classe A.

b. Implémentez la fonction «fnc» de la classe B de manière à ce qu'elle retourne la somme de « i + B::b1 + A::a2 »



2. Implémentez les macros «offset_of» et «container_of» permettant d'obtenir la référence sur l'objet dérivé à partir de la référence sur la classe de base.

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

Microprocesseurs 1 & 2: Travail écrit no 4.

Problème n° 2 (Toolchain)

1. Concevez un Makefile pour la génération de l'application « `exec` », laquelle est composée de 3 fichiers (`file1.c`, `file2.c` et `file3.c`). Pour la génération de l'application on utilisera le compilateur GNU « `gcc` » avec les flags « `-g -Wall -Wextra -O2 -std=c99 -MD` ». Pour rappel, le flag « `-MD` » permet de générer les dépendances. Le Makefile devra également permettre d'effacer les fichiers générés pour une cible donnée. Il est impératif d'utiliser des variables pour spécifier les flags de compilation et les fichiers sources. La génération des codes objets sera faite à l'aide d'une règle.

Makefile :

2. Indiquez la fonction des 2 de ces 4 utilitaires suivants :

a. `gcov` :

b. `objdump` :

c. `strip` :

d. `gprof` :

3. Indiquez en une phrase la méthode pour déboguer une application fonctionnant sur une cible à partir d'une machine hôte.

Microprocesseurs 1 & 2: Travail écrit no 4.

Problème n° 3 (Vérification)

1. Citez 2 techniques/méthodes permettant de valider des applications logicielles dans les différentes phases de leur développement
2. Décrivez une technique/méthode permettant de garantir qu'un composant logiciel a été correctement et si possible complètement vérifié. Citez un utilitaire de la chaîne d'outils GNU permettant de mettre d'utiliser cette méthode/technique ainsi que la façon de le mettre en œuvre.
3. Décrivez succinctement le concept de revues de construction
4. Implémentez un test unitaire permettant de valider/vérifier deux résultats positifs et un résultat négatif pour la fonction « `strrchr()` » de la librairie standard C (selon description ci-dessous).

```
/** string scanning operation
 *
 * The strrchr() function shall locate the last occurrence of c (converted to a char)
 * in the string pointed to by s. The terminating null byte is considered to be part
 * of the string.
 *
 * @return Upon successful completion, strrchr() shall return a pointer to the byte
 *         or a null pointer if c does not occur in the string.
 */
char *strrchr(const char *s, int c);
```

Microprocesseurs 1 & 2: Travail écrit no 4.

Problème n° 4 (Documentation)

1. Citez 4 outils permettant de simplifier le développement de logiciels et d'améliorer sa qualité
2. Citez les 3 niveaux principaux de la documentation du logiciel (*public cible*)
3. Décrivez succinctement l'utilité d'un SCM (Source Code Management Tool) tel que GIT ou SVN
4. Indiquez une manière de structurer le logiciel et sa documentation afin de simplifier son développement et sa maintenance

Microprocesseurs 1 & 2: Travail écrit no 4.

Problème n° 5 (Mémoire cache et MMU)

1. Décrivez succinctement le principe d'un DMA.
2. Indiquez à l'aide d'un graphique les 4 phases principales d'un transfert DMA
3. Décrivez succinctement la fonction de la mémoire cache et citez les deux principes qui sont à son origine.
4. Citez deux algorithmes de remplacement de ligne dans la mémoire cache
5. Décrivez succinctement les deux algorithmes d'écriture des données dans la mémoire cache