

Grp
☐A
☐B
☐C

Nom

Contrôle court n°3

Calculatrice et documents interdits - Durée 1 heure - Répondre sur la feuille

1. QUESTIONS DE COURS : GENERALITES SUR LA COMPILATION

1.1. Quels sont les différents moyens de faire exécuter un programme à un ordinateur ?

Illustrez votre réponse par les différents langages que vous avez rencontré (Assembleur, C, JavaScript, Java, Matlab, Python, Shell Unix...). Précisez les logiciels nécessaires à l'exécution de ces programmes et les avantages ou les inconvénients de ces techniques.

1.2. Donnez les étapes pour passer d'un algorithme à l'exécution d'un programme (pensez aux éventuelles différences selon les langages et système d'exploitation Linux ou DOS) en donnant quelques exemples d'opérations effectuées ainsi que les types de fichiers créés.

2. ANALYSE DE TEXTE

Voici un article publié dans PC Expert du mois d'avril.

Le règne du BIOS touche à sa fin

Déjà en place sur les stations équipées de processeur Itanium, l'Extensible Firmware Interface (EFI), le successeur présumé du Bios, ne devrait pas tarder à débarquer sur les PC de bureau. Microsystème d'exploitation, l'EFI est capable d'apporter des innovations au moment du démarrage du PC. Il supporte l'affichage en 800 x 600 x 32bits et gère de nombreuses options au démarrage (Ethernet, USB...). De plus, la programmation est réalisée en C interprété, ce qui devrait accélérer, sécuriser et simplifier le développement des mises à jour.

Quel(s) avantage(s) et quel(s) inconvénient(s) voyez vous à utiliser du C interprété plutôt que de programmer en assembleur compilé comme c'était le cas avant ? Expliquez très succinctement.

3. ETUDE D'UN PROGRAMME COMPILE

On a récupéré sous forme de fichier objet une fonction dont le code avait été écrit en C.

3.1. Pourquoi ne peut-on pas directement lire le contenu de ce fichier ?

3.2. On aimerait utiliser cette fonction dans d'autres fichiers.

De quel autre fichier devrait-on disposer (à part le fichier source) pour y parvenir facilement ?

Que doit-on faire avec ce fichier ? Et avec le fichier objet ?

3.3. On essaye de reconstituer le prototype de la fonction en désassemblant le fichier objet.

Avant d'en étudier le code ci-dessous, rappelez les règles générales de compilation du C pour :

- le passage de paramètres

- le retour de valeur

3.4. Voici le code récupéré à partir du fichier objet :

```
@1@0:  push    bp
        mov     bp,sp
@1@1:  sub     sp,2
        mov     word ptr [bp-2],2      ;word = 16 bits
@1@2:  jmp     short @1@7

@1@3:  mov     ax,word ptr [bp+4]
        cdq
        idiv    word ptr [bp-2]        ;préparation pour la division entière
        ;division de ax par le paramètre
@1@4:  cmp     dx,0                     ;dx contient le reste
        jne     short @1@6
@1@5:  mov     ax,0
        jmp     short @1@A
@1@6:  inc     word ptr [bp-2]
@1@7:  mov     ax,word ptr [bp-2]
        imul    ax,word ptr [bp-2]      ;multiplication entière
@1@8:  cmp     ax,word ptr [bp+4]
        jl      short @1@3
@1@9:  mov     ax,1
        jmp     short @1@A
@1@A:  mov     sp,bp
        pop     bp
        ret
```

Entre @1@3 et @1@A, qu'y a-t-il dans la pile en :

- BP-2 ?

- BP ?

- BP+2 ?

- BP+4 ?

En déduire le prototype de la fonction.

Que fait-elle à votre avis ?