Université BORDEAUX I

Année 1996-97
Session de janvier

Maîtrise d'informatique

Modèles et Techniques de Programmation (J.P.Braquelaire)

Durée: 2 heures
Tous documents autorisés

Première partie

1) – 2 points – Donner en Scheme un exemple d'appel récursif terminal et un exemple d'appel récursif non terminal (chaque exemple n'excèdera pas cinq lignes). Expliquez brièvement votre réponse.

2) – 1 point – Donner un exemple d'appel récursif terminal en C (l'exemple n'excèdera pas cinq lignes). Expliquez brièvement votre réponse.

3) – 1 point – Expliquer la différence entre la gestion de l'appel récursif terminal en C et en Scheme.

4) – 1 point – Quel(s) langage(s) connaissez-vous permettant la programmation CPS. Argumentez votre réponse.

Deuxième partie

Sous UNIX, la notification d'une erreur lors de l'exécution d'un appel système s'effectue en retournant la valeur -1.

5) – 3 points – Proposer une solution raisonnable[1] pour améliorer ce mécanisme. On développera un exemple en traitant une fonction d'interface pour un appel système au choix.

Troisième partie

On considère des images numériques rectangulaires et monochromes (chaque point de l'image est affiché avec une intensité de gris). On suppose que l'entier 0 code le noir, et l'entier `intensite_maximum` le blanc. Les intensités de gris sont codées par des entiers compris entre 0 et `intensite_maximum`. La quantité `intensite_maximum` dépend de chaque image (c'est le nombre de niveaux de gris sur lequel est codé l'image).

6) – 3 points – Donner l'interface en C ANSI du module `image`. On supposera que ce module contient seulement quelques fonctions d'interface élémentaires : `creer`, `liberer`, `largeur_image`, `hauteur_image`, `intensite_maximum`, `lire_pixel`, `ecrire_pixel`.

T.S.V.P.

[1] Une modification du noyau Unix n'est pas considérée comme une solution raisonnable.

Une *convolution discrète* est une opération calculant une image I' à partir d'une image I et d'une matrice K de nombres réels, en général carrée et de petite taille par rapport à l'image. On suppose que la matrice K est de taille $(2p+1)\times(2p+1)$ et on note ses éléments $K_{i,j}$, $-p \leq i \leq p$, $-p \leq j \leq p$. Sous ces hypothèses, la valeur du point $I'[x][y]$ de la nouvelle image est définie par

$$\sum_{i=-p}^{p} \sum_{j=-p}^{p} I[x+i][y+j]\, K_{i,j}$$

7) – 4 points – Écrire la fonction C qui construit une image I' à partir d'une image I codée par le module `image` de la question précédente et d'une matrice K connue explicitement (on supposera que la fonction `lire_pixel` du module `image` retourne une valeur cohérente, par exemple 0, pour une position située au dehors de l'image).

Cette solution n'est pas très efficace car il y a un grand nombre d'accès aux pixels de l'image I et que chaque accès nécessite un appel de fonction. Pourtant, un accès direct aux pixels de l'image soulève un problème d'exportation d'implémentation.

8) – 5 points – Discuter et comparer, du point de vue de la modularité, différentes possibilités d'améliorer l'efficacité de ce programme. Laquelle préconiseriez-vous ?

Remarque. Il n'y a pas nécessairement **une** meilleure solution. L'important est de fournir une argumentation convainquante.