

Examen de Concepts et Programmation Système (HLIN504) Session 2

Michel Meynard

Durée : 2 heures

UNIQUE DOCUMENT AUTORISÉ : "Aide mémoire Unix et librairie standard C v2.0"

26 janvier 2015

1 Compréhension du cours

Répondez à ces questions concisément. Une réponse fautive sera comptée négativement !

Question 1 En C, que signifient les 3 mots suivants : NULL, '\0', "" ? Exprimez leur valeur en hexadécimal.

Question 2 Comment comparer deux chaînes de caractères en C ? Donnez 3 exemples significatifs.

Question 3 Comment s'appelle la table d'un système de fichiers qui emmagasine l'information relative à tous les fichiers ? Décrivez succinctement 6 champs d'une entrée de cette table. Comment accéder à ces champs en programmation C (appel système) ?

Question 4 Un signal envoyé à un processus contient-il l'identification du processus émetteur ? Comment gérer une temporisation sur l'entrée d'un utilisateur un peu lent (donner un exemple d'attente de 5 secondes maxi pour un scanf) ?

2 Tri et comptage de mots

On souhaite trier en ordre ascendant et compter les **mots** d'un fichier texte. Pour simplifier, on considérera qu'un mot est constitué d'une séquence de lettres minuscules comprises entre a et z. Tout autre caractère (espaces, retour lignes, majuscules, lettres accentuées, ...) ou suite de caractères sera vu comme un séparateur de mots et ne sera pas pris en compte. **Le fichier n'est pas borné donc on ne peut pas le copier intégralement en mémoire centrale ni copier tous ses mots. Toute solution copiant tous les mots dans un tableau puis triant ce tableau seront considérées comme nulles !**

Un exemple d'un tel fichier `exemple.txt` est donné ci-dessous :

```
mon ami leo va en ville. puis,  
leo s'en va !
```

On souhaite écrire un programme C affichant la liste triée en ordre croissant des mots et de leur nombre d'occurrences d'un tel fichier dont le nom est passé en paramètre à la ligne de commande :

```
$tript exemple.txt  
ami (1), en (1), leo (2), mon (1), puis(1), s (1), sur (1), va(2), ville(1)  
$
```

2.1 Travail

Afin de décomposer le travail, on souhaite écrire une fonction `lireMot` dont la spécification suit :

```
/** lecture séquentielle du fichier (déjà ouvert), en ignorant ce qui n'est pas une lettre minuscules, et  
 * retourne dans le buffer une copie du premier mot trouvé sous forme d'une chaîne C.  
 *  
 * @param f fichier déjà ouvert en lecture  
 * @param buf un tampon de caractères (déjà alloué) qui contiendra le mot lu au retour  
 * @return 0 si EOF, 1 si trouvé un mot (qui est dans buf)  
 */
```

Question 5 Écrire l'algorithme de la fonction `lireMot`

Question 6 Écrire la fonction C `lireMot` correspondant à cet algorithme **EN UTILISANT DES APPELS SYSTÈMES**.

Selon le modèle du crible d'Eratosthène parallèle, on souhaite paralléliser le programme `tript` en utilisant des processus et des tubes anonymes. Chaque processus créé (fonction `triptpar`) récupérera dans son tube d'entrée la liste des mots (chaînes), sélectionnera et affichera le plus petit ainsi que son nombre d'occurrences tout en transmettant les autres via un tube de sortie à un processus enfant qu'il aura créé. La fonction principale (`main`) n'aura qu'à créer le premier tube et le premier fils, puis lira séquentiellement les mots dans le fichier et les enverra dans le tube.

Question 7 Dessiner un schéma décrivant les processus créés ainsi que les tubes et leur contenu en utilisant l'exemple de texte suivant : `cd baba abc baba`.

Question 8 Ecrire l'algorithme de la fonction `main()`.

Question 9 Ecrire l'algorithme de la fonction `trictpar(int in[2])`.

Question 10 Ecrire le programme C `tript.c`.