Licence d'informatique
Module de C/Unix

Projet C

`t9` — Écrire avec 9 touches (et un doigt !)

Jean-Luc LEVAIRE

Décembre 1998

Les organiseurs ne disposent pas de clavier alpha-numérique en raison de leur petite taille. Ils utilisent à la place un clavier affiché sur un écran tactile, et l'utilisateur sélectionne une touche affichée sur cet écran à l'aide d'un stylet. L'inconvénient majeur d'une telle interface est que les touches doivent rester minuscules pour conserver une surface d'écriture suffisante. Pour remédier à ce défaut, les constructeurs ont eu l'idée de grouper plusieurs lettres sur une seule touche. Le processeur de l'organiseur a alors la charge de proposer une liste de mots possibles, qui correspondent à la séquence de touches sélectionnées. De plus ces mots sont proposés dans l'ordre de leur fréquence d'apparition la plus élevée dans les textes. Le premier mot de cette liste sera alors le mot affiché par défaut. Cette technologie a été développée par la société Tegic (`www.tegic.com`) sous le nom de T9 (Type with nine keys). Elle est en cours d'implémentation sur les téléphones portables, afin de pouvoir envoyer des courriers électroniques, et non plus seulement en recevoir. L'association des touches avec les lettres est donnée Fig. 1.

Le processeur de l'organiseur utilise, pour sélectionner les mots possibles du vocable, un dictionnaire contenant une liste de mots associés à leur fréquence d'apparition dans les textes. Ce dictionnaire est stocké dans un tableau `vocable` de structures contenant un mot suivi de sa fréquence d'apparition.

La fréquence d'apparition `freq` est un entier représentant une fréquence relative. Les mots sont classés par longueur dans le tableau `vocable`, et pour une longueur de mot donnée, ils sont classés dans l'ordre lexicographique. Pour accélérer la recherche, un tableau `lstart` indique le début des zones de `vocable` correspondant à des mots de longueur donnée, ainsi que le nombre de mots de cette longueur. Ce tableau comporte `MAXLENGTH` entrées. La figure 1 présente schématiquement cette structure de données.

En plus des mots de ce dictionnaire, l'utilisateur peut ajouter des mots de son propre vocabulaire dans le dictionnaire. Pour stocker ces nouveaux mots ainsi que leurs fréquences, on utilise pour chaque longueur de mots, une liste des mots définis par l'utilisateur. On suppose que les mots ajoutés ont moins de `MAXLENGTH` caractères. Le début de cette liste est donnée par le champ `new`.

Exercice 1

Écrire un module `accesdico` de chargement et de sauvegarde du dictionnaire sur un fichier `dico.cmp`. Le chargement s'effectuera en mémoire dans un tableau alloué par le module. Le fichier sera compressé. La compression est réalisée par l'élimination des préfixes communs des mots et par leur remplacement par un caractère unique qui code le nombre de caractères communs avec le mot précédent. Chaque mot est suivi de sa fréquence relative.

Voici un exemple de dictionnaire et de sa version compressée :

Id : t9.tex,v 1.1 2000/11/19 09 :56 :19 marquet Exp

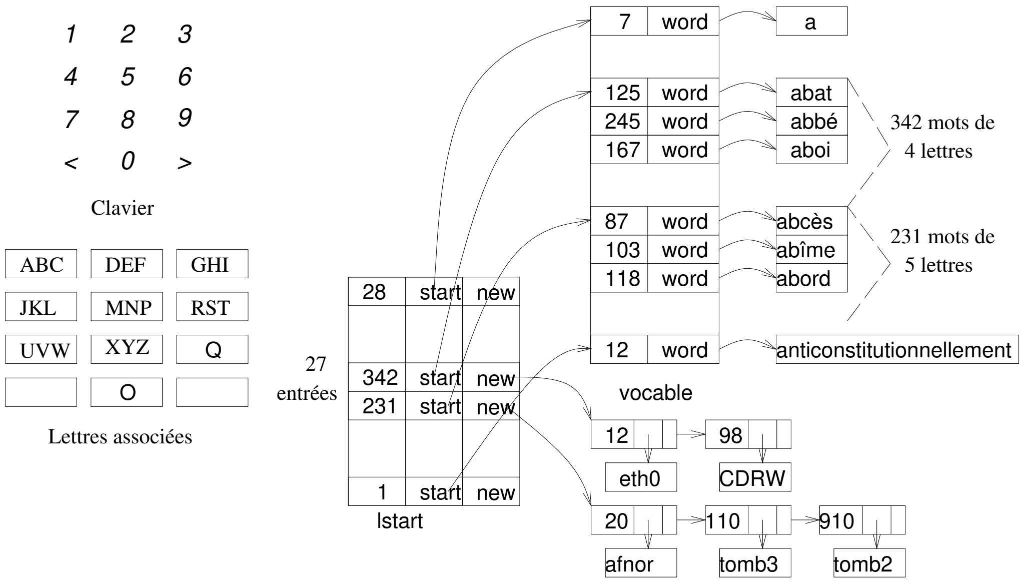

FIG. 1 – Le clavier T9 et la structure de données proposée.

```
% cat /tmp/dico
computer 123
computerize 122
computerized 121
computers 145
inform 156
informal 13
informality 12
```

```
% cat /tmp/dico.cmp
0computer 123
8ize 122
Bd 121
8s 145
0inform 156
6al 13
8ity 12
```

Les caractères de codage sont les suivants :

0-9 représentent les entiers de 0 à 9 ;

A-Z représentent les entiers de 10 à 35 ;

a-z représentent les entiers de 36 à 61.

Exercice 2
Écrire un module `trouvermot` qui recherche dans le dictionnaire les mots correspondant à la séquence de touches tapées jusqu'à présent. Les mots sont regroupés en une liste que l'on trie par ordre décroissant de fréquence d'apparition. La recherche des mots ne tient pas compte des lettres tapées par l'utilisateur. Les lettres ne sont utilisées que pour l'insertion d'un mot nouveau. Il ne doit pas y avoir de recopie des mots du dictionnaire, la liste devant pointer directement sur les mots du dictionnaire. Cette liste est utilisée en combinaison avec les touches < et >, qui permettent respectivement de passer au mot suivant le plus fréquent, et au mot suivant le moins fréquent, et ce de manière circulaire. L'affichage du mot courant se fait par l'intermédiaire de la variable `current_word` et de la fonction `display_word`. L'affichage des touches frappées se fait par l'intermédiaire de la variable `current_stroke` et de la fonction `display_stroke`. Ce module gérera enfin également la fonction `clear` qui permet de saisir un mot nouveau.

Exercice 3
Écrire la fonction

```
void insert(void) ;
```

qui insère dans le dictionaire le mot nouveau présent dans la fenêtre “What you strike”, avec une fréquence d'apparition de valeur 1. Ce mot est contenu dans la variable `current_stroke` (ce que vous devez gérer). Cette fonction est appelée lorsque le bouton Insert est appuyé.

Exercice 4
Afin de personnaliser encore l'organiseur, il est nécessaire de modifier les fréquences d'apparition des mots suivant ce que tape réellement l'utilisateur. Il suffit d'incrémenter la fréquence de chaque mot choisi lors des `clear`. Les fréquences deviennent alors absolues. Il faut une procédure pour les rendre relatives. Soit N la somme des fréquences des mots à la précédente relativisation, n le nombre de mots entrés depuis cette dernière relativisation. Pour chaque fréquence x d'un mot, la nouvelle fréquence sera $\left\lceil \frac{N \times x}{N+n} \right\rceil$. Ecrire un module qui relativise les fréquences du dictionnaire. Ce module sera appelé lors de la sauvegarde du dictionnaire, ou lorsqu'une fréquence devient trop grande.

Mise en œuvre

Je vous ai écrit une interface graphique représentant un clavier avec deux fenêtres, une pour le mot suggéré par le dictionnaire, l'autre pour les touches réellement tapées au clavier. Cette interface contient la fonction `main`, qui est une boucle inifinie gérant les événements souris et clavier (entre autres). Vous devez essentiellement intervenir au niveau de la fonction `HandleTouch` qui est appelée lors d'un clic souris sur une touche du clavier. De même pour les fonctions `insert`, `clear` et `quit`, appelées lors d'un clic sur le bouton correspondant. Vous trouverez tout ceci dans `~levaire/TPC/sample/projet9899/`