Licence d'informatique
Module de C/Unix

Examen – Partie C

Jean-Luc LEVAIRE

Janvier 1998

Les documents de cours et TD sont autorisés.

On rendra deux copies séparées pour la partie Unix et pour la partie langage C. La partie C est à rendre sur une *copie de couleur blanche*.

1 Questions de cours

Question 1
Qu'affiche le programme suivant ?

```
void
f(int **i)
{
  (*(*++i)--)++;
}

void
main(void)
{
  int i[] = {0, 2, 4, 6}, *p[2] = {i, i+2};
  int j;
  f(p);
  for (j=0; j < 4; j++)
    printf("%d ", *(i + j));
  printf("\n%d %d\n", **p, **(p + 1));
}
```

Question 2
Au niveau du système de fichier, quelle est la différence entre un lien physique et un lien symbolique ?

Question 3
Que signifient les déclarations suivantes :

- `char *(*(*d1)[])(int);`
- `int (*d2(FILE *(*d3)(int)))(FILE *);`

2 Écrire avec 9 touches (et un doigt !)

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.

Id : xamc98.tex,v 1.2 2000/11/18 09 :41 :32 marquet Exp

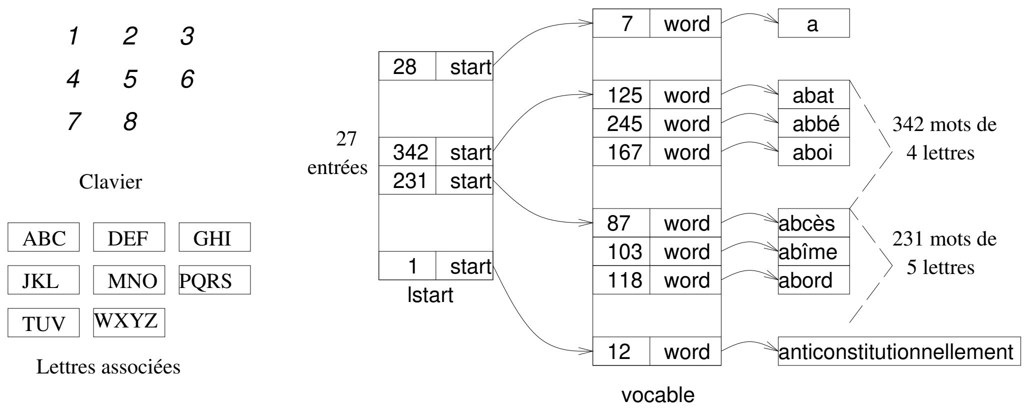

FIG. 1 – Le clavier T9 et la structure de données utilisée.

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 (http ://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 (8 touches sont en réalité utilisées).

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 :

```
#define NBWORDS 60000
typedef struct entry {
  char *word;
  int freq;
} entry, vocable[NBWORDS];
```

La fréquence d'apparition `freq` est un entier représentant une fréquence relative, `NBWORDS` est le nombre de mots du dictionnaire. 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 la structure de données complète.

Question 4
Définissez un type `lengthstart` pour un élément du tableau `lstart`, et donnez la définition de ce tableau.

Question 5
Écrire la fonction

```
entry *searchstart(char *seq, lengthstart lstart[]) ;
```

qui retourne un pointeur sur la zone des mots dont la longueur correspond à la séquence de touches `seq`. La séquence de touches `seq` est représentée par une chaîne de caractères contenant la suite des chiffres de la séquence (¨`1312`¨ par exemple).

Connaissant la zone de recherche, il faut maintenant déterminer la liste des mots (avec leur fréquence) qui correspondent réellement à la séquence de touches frappées, et enfin trier cette liste par fréquence d'apparition des mots.

Question 6
Définissez un type `wordlist` pour représenter une liste de pointeurs sur des mots associés à leur fréquence.

Question 7
Écrire la fonction

```
int match(char *seq, char *word) ;
```

qui retourne `1` si le mot `word` correspond à la séquence de touches `seq`, et `0` sinon.

Question 8
Écrire la fonction

```
wordlist searchword(char *seq) ;
```

qui retourne la liste des mots possibles (avec leur fréquence) correspondant à la séquence `seq`. Pour minimiser l'espace utilisé, cette liste pointera directement dans le tableau `vocable`.

Question 9
Le type `words` utilisé pour stocker l'ensemble des mots possibles associés à une séquence de touches est le suivant :

```
typedef struct wordarray {
  int number;
  char **array;
} wordarray, *words;
```

`number` est le nombre de mots dans l'ensemble, et `array` est un tableau de chaînes de caractères contenant chacun un mot. Ecrire la fonction

```
words sortword(wordlist list) ;
```

qui trie la liste de mots par ordre décroissant de leur fréquence d'apparition, et retourne l'ensemble trié des mots possibles. Le tableau de chaînes retourné pointera directement dans le dictionnaire (il n'y a pas copie des mots). De plus cette fonction détruira la liste `list` de mots possibles.