

Projet de Systèmes d'exploitation

2013 – Licence 2

Le but du projet est de réaliser un clone du jeu typespeed (<http://typespeed.sourceforge.net>). Le but du jeu est de taper rapidement les mots qui apparaissent à l'écran.

Le jeu se déroule dans le terminal. La fenêtre se découpe en trois zones. La **zone principale** qui occupe toute la fenêtre sauf la ligne du bas constitue la **zone d'entrée** et la ligne du haut qui constitue la **zone de score**. Les trois zones sont séparées par une ligne constituée du caractère - (moins du 6).

Travail demandé

Le programme est constitué d'un thread effectuant les deux actions suivantes :

- (a) afficher régulièrement des mots dans la zone principale. Le jeu se termine quand il n'y a plus de place dans la zone ;
- (b) afficher ce que l'utilisateur tape au clavier dans la zone d'entrée (il faut que la touche d'effacement fonctionne comme d'habitude). Quand l'utilisateur valide l'entrée en appuyant sur entrée, si le mot est présent parmi les mots affichés, ce dernier doit disparaître et sa longueur s'ajouter aux points affichés dans la zone de score. Si le mot n'est pas présent, la zone d'entrée est effacée.

Il n'y a pas de contrainte particulière sur l'affichage des mots. On pourra se contenter d'un mot par ligne et d'arrêter le jeu quand toutes les lignes sont occupées. Vous êtes libre, **une fois le jeu fonctionnel**, d'améliorer comme il vous semble cet affichage.

Modalité de rendu et de réalisation

Le projet est à réaliser en C et sans utilisation de la bibliothèque d'entrée/sorties standard (stdio.h), sauf les fonctions **sprintf**, **snprintf** et **perror**. On pourra aussi utiliser stderr à des fins de débogage.

Il est à faire en binôme ou tout seul (mais sans contre-partie). Avant la date limite, une archive .zip ou .rar devra être déposée sur EPREL respectant les éléments suivants :

- l'archive devra se nommer Nom1_Prénom1_et_Nom2_Prénom2 avec les noms dans l'ordre alphabétique. Par exemple, le binôme Jean Martin et Sylvie Dupont rendra un archive Sylvie_Dupont_et_Jean_Martin.zip (ou .rar)
- seul l'un des membres du binôme doit déposer l'archive
- l'archive doit contenir :

- un fichier README (format texte) contenant les noms et prénoms des membres du binôme, la manière d'exécuter le programme, les fonctionnalités demandées qui fonctionnent, les améliorations possibles et les bugs connus (un bug inconnu sera plus sévèrement sanctionné qu'un bug connu)
- un fichier Makefile permettant de compiler votre programme **sans Warning avec l'option -Wall**
- les fichiers sources .c (et éventuellement .h)
- les dépôts en retard seront interdits et sanctionnés d'un 0

Exemple de fenêtre d'affichage

Score: 234

licence
informatique
octet

licenz

Indications techniques

Le programme contiendra un seul thread. Ce thread effectuera essentiellement un **select** pour attendre des entrées de l'utilisateur et effectuer l'action (b) avec un timeout permettant d'effectuer l'action (a).

Affichage

Pour éviter des problèmes d'affichage, il faudra mettre le terminal en mode raw avec les fonction **cfmakeraw** et **tcsetattr**. Dans ce mode, les caractères tapés ne sont pas affichés. C'est à votre programme de les afficher. Afin d'éviter de ne plus pouvoir utiliser le terminal une fois le programme terminé, il faudra, avant la mise en mode raw, sauvegarder son état de la manière décrite ci-dessous avec les fonctions **tcgetattr** et **tcsetattr**. L'utilisation de ces fonctions est illustrée par le code suivant :

```
...
struct termios saved_termio;
if (tcgetattr(STDIN_FILENO,&saved_termio)==-1) {
    perror("tcgetattr");
    ...
}
struct termios raw_termio;
memcpy(&raw_termio,&saved_termio,sizeof(struct termios));
cfmakeraw(&raw_termio);
if (tcsetattr(STDIN_FILENO,TCSANOW,&raw_termio)==-1) {
    perror("tcsetattr");
    ...
}
```

```

}
...
... /* programme principal */
...
if (tcsetattr(STDIN_FILENO, TCSANOW, &saved_termio) == -1) {
    perror("tcsetattr");
    ...
}
...

```

Pour se déplacer dans le terminal, on utilisera :

-on déclarera la fonction suivante :

```

int getNumEnv(char *name, int def) {
    char *val=getenv(name);
    if (val!=NULL && *val != '\0') {
        char *end;
        int ans=strtol(val,&end,10);
        if (*end=='\0')
            return ans;
    }
    fprintf(stderr,"%s incorrect, défaut=%d\n",name,def);
    return def;
}

```

-pour avoir le nombre de colonnes du terminal, on utilisera
getNumEnv("COLUMNS",80)

-pour avoir le nombre de lignes du terminal, on utilisera
getNumEnv("LINES",24)

-en affichant (avec **write**) la chaîne obtenue avec
sprintf("\033[%d;%dH",x,y), on se déplace aux coordonnées x,y du
terminal

-si les variables COLUMNS et LINES ne sont pas définies, vous
pouvez exécuter la commande :

```
eval $(resize)
```

dans le terminal avant de lancer le programme (une seule fois par
terminal suffit)

Dictionnaire

Dans un premier temps, les mots seront mis dans un tableau dans le code du programme. Une fois ce dernier fonctionnel, les mots devront être chargés depuis un fichier. Un fichier contenant des mots est disponible sur EPREL. Il est demandé pour le projet de charger le dictionnaire à l'aide d'un **mmap** sur un fichier à partir de celui fourni. Il faudra ajouter un index au début du fichier afin de ne pas avoir à chercher les mots. Ainsi, les quatre premiers octets du fichier contiendront un int indiquant le nombre de mots. Les int suivants seront les octets du fichiers auxquels se trouvent les mots et enfin viendront les mots eux-même terminés par un '\0'. Par exemple, si les mots sont licence, info et octet, le fichier contiendra (octet par octet, caractères entre apostrophes, les int sont sur quatre octets) :

```
3 0 0 0 16 0 0 0 24 0 0 0 29 0 0 0 'l' 'i' 'c' 'e' 'n' 'c' 'e' 0
'i' 'n' 'f' 'o' 0 'o' 'c' 't' 'e' 't' 0
```

Le 'l' de licence est le 16^e octet du fichier (sachant que l'on commence à 0), le 'i' de info est le 24^e et le 'o' de octet est le 29^e.

Ainsi, faire un **mmap** sur le fichier et un cast du pointeur obtenu vers un point sur la structure :

```
struct dict {
    int count;
    char *words[];
}
```

doit permettre d'obtenir le dictionnaire.