

INF203 - Exercices semaine 12

Interpréteur de commandes (suite)

Exercice 1 :

Dessinez un automate permettant de gérer les boucles `while` dans notre interpréteur de commandes.

Exercice 2 :

Nous souhaitons écrire un ensemble de fonctions permettant de gérer un historique des lignes de commandes et permettant de se déplacer dans cet historique. Nous disposons de la fonction de prototype :

```
int lire_ligne_fichier(FILE * fichier, char *ligne);
```

écrite au précédent TP, qui lit une ligne depuis le flux de descripteur donné et renvoie 1 si une ligne a été lue et 0 sinon (fin de fichier et aucune ligne lue). Nous supposons que l'ensemble des informations concernant notre historique sont stockées dans les variables globales suivantes :

```
char lignes_lues[NOMBRE_MAX_LIGNES][TAILLE_MAX_LIGNE];  
int nombre_lignes_lues;  
int position_courante;  
FILE *fichier;
```

1. écrivez une fonction de prototype :

```
void init_historique(FILE * f);
```

permettant d'initialiser le nombre de lignes lues et la position courante à 0 ainsi que le descripteur de fichier de l'historique à celui passé en paramètre.

2. écrivez une fonction de prototype :

```
int lire_ligne(char *ligne);
```

qui, selon la valeur de la position courante et du nombre de lignes lues :

- lit une ligne depuis le flux d'entrée, la copie dans l'historique et dans la ligne à lire
- copie une ligne depuis l'historique

Cette fonction devra également faire avancer la position courante et le nombre de lignes lues (le cas échéant). Elle devra renvoyer la valeur renvoyée par `lire_ligne_fichier` si cette dernière est appelée et 1 sinon.

3. écrivez deux fonctions permettant de se déplacer dans l'historique, de prototypes respectifs :

```
— int obtenir_numero_ligne();
```

qui permet de récupérer le numéro de la dernière ligne lue (attention, normalement la position courante est le numéro de la prochaine ligne à lire).

```
— void aller_a_la_ligne(int numero);
```

qui permet de changer la valeur de la position courante.

Exercice 3 :

Comment vous y prendriez vous pour gérer les boucles `for`? Quelles seraient les différences avec les boucles `while`?

Exercice 4 :

Utilisez tout le temps restant (s'il en reste) pour clore les TD de l'UE :

- réponses aux questions ouvertes
- retour sur les exercices passés