

ZZ1 – C – TP n°4

Tableau de structures / Liste chaînée simple

Le but de ce TP est de gérer les informations d'un utilisateur Unix sur plusieurs machines.

1) Utilisation d'un tableau de structures

Soit la structure `login_t` qui stocke les informations présentant un champ `nom_de_login` sur 8 caractères, un champ `mot_de_passe` sur 8 caractères, un champ indiquant le `nom_de_la_machine` pour laquelle ces informations sont valides. On pourra aussi ajouter un nombre entre 0 et 10 : 0 la machine est souvent utilisée, 10 elle n'est quasiment jamais utilisée.

Ecrire un programme manipulant un tableau de structures `login_t`. Le tableau sera statique ou dynamique. Les deux fonctions suivantes sont à définir en particulier.

- a. Saisie sur un élément du tableau
- b. Affichage du tableau

2) Utilisation d'une liste chaînée

Ajouter un champ pointant sur la structure elle-même (pour chaîner ces informations). Ecrire un programme capable de gérer une liste chaînée d'informations.

- a. Choisir la définition de la tête de liste (fictive ou réelle) et initialiser la liste
- b. Insérer en tête un premier élément et l'afficher.
- c. Insérer un deuxième élément et l'afficher
- d. Insérer un troisième élément et l'afficher
- e. Faire une boucle d'affichage dans la liste
- f. La déplacer pour en faire une fonction
- g. Créer une fonction pour insérer un élément en tête de liste
- h. Écrire une fonction pour libérer **toute** la mémoire. Cette dernière étape devra être validée par des tests et l'usage de **valgrind**.

Pour ceux qui découvrent le C cette année et qui ont d'abord utilisé une tête fictive, il est conseillé de refaire cette partie avec une tête de liste réelle.

Les contraintes sur les mots de passe sont les suivantes : 8 caractères minimums, au moins un caractère en majuscule, au moins un chiffre ou un caractère différent des lettres de l'alphabet. Pour votre TP les mots de passes sont stockés « en clair ». Développez vos propres fonctions pour tester les caractères valides, cependant regardez les fonctions disponibles dans `<ctype.h>`

Pour les DUT qui s'ennuient : écrire également une fonction de génération automatique de mots de passe. Lors de l'ajout d'un utilisateur, il sera ainsi possible de proposer un mot de passe 'correct' à l'utilisateur. Les développeurs plus expérimentés peuvent utiliser la fonction de cryptage d'Unix.

CONSEILS :

- **Procéder par étapes, ajouter les fonctions une à une et les valider également au fur et à mesure**
- **Commenter et aérer votre code source en vous inspirant du guide de style fourni par l'ISIMA**