

Boumediene Boukharouba
Alexandre Deneault
IFT2035 Concepts des langages de programmation

TP1 GESTION MÉMOIRE ET POINTEURS
Rapport

Travail présenté à
Monsieur Marc Feeley

Université de Montréal
17 octobre 2016

1. Fonctionnement général du programme

Le programme représente une calculatrice à précision infinie. Il demande à l'utilisateur d'entrer une ligne de texte décrivant un calcul au format postfixe. Par la suite, le programme découpe la ligne en mots. En lisant le premier caractère d'un mot, le programme détermine à quelle catégorie il appartient. (opérateur, opérande, variable, assignation de variable) Il sait qu'un mot est terminé lorsqu'il rencontre un espace. Pour chaque mot, le programme exécute les opérations nécessaire.

Lorsque le mot est un nombre, le programme l'enregistre dans une liste chaînée qu'il ajoute sur le dessus de la pile. S'il s'agit d'une variable à laquelle un nombre a été assignée, il empile ce nombre. Si le mot est un opérateur, le programme dépile le bon nombre d'opérande pour effectuer l'opération et empile le résultat. Si le mot correspond à une assignation de variable, le programme va chercher le pointeur qui se trouve sur le dessus de la pile pour l'assigner à cette variable.

Les calculs se font au fur et à mesure afin de pouvoir libérer la mémoire occupé par les opérandes le plus tôt possible. Ainsi, lorsque l'analyse de la ligne est terminée, il ne reste que le résultat des opérations dans la pile. Le programme le dépile et le convertit en chaîne de caractères avant de l'afficher et de le vider de la mémoire.

Si le programme détecte une erreur, il va vider le buffer d'entrée et la pile et rétablir les nombres qui se trouvaient dans les variables avant la ligne qui a causée l'erreur.

2. Problèmes de programmation

a) Nombres et variables

Les nombres sont représentés par une structure comportant un pointeur vers la première cellule d'une liste chaînée de caractère, un entier pour le signe et un entier pour le compteur de référence. Chaque cellule de la liste chaînée contient une union pouvant contenir un caractère représentant le chiffre et un pointeur vers la prochaine cellule. Le chiffre de poids le plus élevé est à la fin de la liste chaînée pour faciliter les calculs.

Les variables sont représentées par un tableau de pointeur. La variable 'a' est à l'indice 0, 'b' est à l'indice 1, 'c' est à l'indice 2, etc. Il y a aussi un deuxième tableau de variables temporaires. Toutes les variables sont initialisées à « NULL ». Lorsqu'on assigne une valeur à une variable, le pointeur sur le nombre est placé dans le tableau de variable temporaire. S'il y a déjà un pointeur dans le tableau temporaire, ce dernier est placé dans une pile de nombres à libérer à la fin de l'analyse de la ligne. Après avoir terminé de lire la ligne, s'il n'y a pas eu d'erreur, le programme transfère les données du tableau temporaire vers le tableau des variables. Dans le cas contraire, le programme vide le tableau temporaire, laissant les variables dans l'état où elles se trouvaient avant l'exécution de la ligne. Dans les deux cas, la pile de nombres à libérer est vidée.

b) Analyse et calcul

Le programme analyse les lignes caractère par caractère. Il utilise le premier caractère de chaque mot pour déterminer à quelle catégorie il appartient (opérande, opérateur, variable ou assignation de variable). Il exécute l'algorithme associé à la catégorie avant de passer au prochain mot.

Si le caractère est un chiffre, le programme crée une structure pour sauvegarder le nombre et l'ajoute sur le dessus de la pile. La pile est représentée par un pointeur vers un nœud qui contient un pointeur vers le deuxième nœud de la pile et dont l'union contient un pointeur vers un nombre. Par la suite, le programme ajoute tous les chiffres suivants au début de la liste chaînée du nombre.

Si le caractère est un opérateur, le programme dépile un ou deux nombres. Il exécute ensuite la procédure correspondant à l'opérateur avant d'empiler le résultat.

Si le caractère est une lettre, le programme vérifie que le nom de la variable est valide et qu'une valeur lui a été assignée. Si c'est le cas, il empile le pointeur qui se trouve à la position correspondant au nom de la variable.

Si le caractère est le symbole d'égalité, le programme extrait le nom de la variable et copie le pointeur qui se trouve sur le dessus de la pile dans la case correspondante du tableau de variable temporaire.

Lorsque l'analyse de la ligne est terminée, le programme dépile le dernier nombre qui reste dans la pile et le converti en chaîne de caractères pour pouvoir l'afficher. L'algorithme de conversion détermine le nombre de case à demander en parcourant la liste à l'aide d'un compteur. Ensuite, il extrait les caractères et les place dans le tableau avant d'inverser l'ordre. Finalement, le programme transfère les modifications du tableau des variable temporaire vers le tableau des variables et vide la pile de nombres à libérer avant de passer à la prochaine ligne.

c) Gestion de la mémoire

Dans le programme, il y a une procédure qui s'occupe de libérer l'espace mémoire occupé par le nombre et tous les nœuds de la liste chaînée si le compteur de référence est égal à 0. Cette procédure est appelée à chaque fois qu'un pointeur n'est plus utile.

L'espace pour les structures représentant les nombres sont demandé lorsque nécessaire, c'est-à-dire lors de la lecture d'un nombre ou lors d'une opération. Dans le cas des opérations, les opérandes sont libérés immédiatement après l'opération. L'espace des nœuds pour la pile est demandée dans la macro d'empilement et libéré dans la macro de dépilement. L'espace pour la chaîne de caractère du résultat est libéré après l'affichage, en même temps que le nombre représentant le résultat. Lorsque le programme modifie un pointeur dans l'un des tableaux de variables, l'ancienne valeur est libérée ou placée sur la pile de variables à libérer s'il y en avait une. S'il y a une erreur dans la ligne

ou s'il manque de place en mémoire, le nombre sur lequel le programme travaillait est libéré s'il n'est pas sur la pile. Ensuite, le programme se rend à la procédure d'erreur qui vide complètement la pile, le tableau temporaire de variable et la pile de variables à libérer en appelant la procédure de libération d'espace pour chaque nombre.

d) Algorithmes

Addition :

Pour effectuer l'addition, le programme commence par dépiler deux nombres de la pile. Il compare ensuite les signes des opérandes. Si les signes sont différents, il en modifie un et effectue une soustraction en inversant l'ordre des opérandes si nécessaire. Il rétablit le signe modifié après la soustraction. Si les signes sont semblables, le programme crée la structure pour mémoriser le résultat. Il extrait ensuite les chiffres caractère par caractère et ajoute le résultat de l'addition à la fin de la liste chaînée du résultat en conservant la retenue dans une variable pour pouvoir l'ajouter à l'addition suivante. Le programme libère ensuite les deux opérandes et empile le résultat.

Soustraction :

Pour effectuer la soustraction, le programme commence par dépiler deux nombres de la pile. Si les signes des deux nombres sont différents, il modifie le signe du deuxième opérande et effectue une addition. Il rétablit le signe modifié après l'addition. Si les signes sont semblables, le programme crée la structure pour mémoriser le résultat et compare les deux nombres. Si le premier est plus grand que le deuxième, le signe du résultat sera le même que celui des opérandes, sinon il inverse le signe et l'ordre des deux nombres. Maintenant que le premier nombre est plus grand que le deuxième, le programme peut extraire les chiffres caractère par caractère et effectuer les soustractions en ajoutant les valeurs dans la liste chaînée de résultat. Si le résultat de l'une des soustractions est plus petit que 0, il ajoute 10 à ce résultat et enlève 1 de la soustraction suivante à l'aide d'une variable, ce qui va toujours être possible puisque le premier nombre est plus grand que le deuxième. Il garde aussi un pointeur vers le dernier résultat différent de 0 pour pouvoir les enlever des chiffres de poids élevés. Il libère ensuite les deux opérandes et empile le résultat.

Multiplication :

Pour effectuer la multiplication, le programme commence par dépiler deux nombres de la pile puis il crée la structure pour accueillir le résultat qui est initialisé à 0. Il mémorise les signes des opérandes dans des variables pour pouvoir les rendre positifs le temps des calculs et les rétablir à la fin de l'opération. Il intervertit les deux nombres si nécessaires pour avoir le plus petit en deuxième. Le programme extrait chacun des chiffres du deuxième nombre. Pour chaque chiffre, il additionne le premier nombre au résultat autant de fois que nécessaire. Il ajoute ensuite un '0' à la fin du premier nombre (multiplication par 10) pour la prochaine itération (en gardant un pointeur sur le premier 0 ajouté afin de pouvoir les retirer par la suite). Lorsque c'est terminé, le programme détermine le signe du résultat à l'aide des signes des deux nombres. Il libère ensuite les deux opérandes et empile le résultat.

e) Traitement des erreurs

Il y a un bloc de traitement des erreurs à la fin de la boucle de lecture des lignes. Il y a aussi une variable globale qui permet d'enregistrer un message d'erreur. Dès que le programme détecte une erreur, que ce soit un pointeur nul à la suite de malloc() parce qu'il n'y a pas assez de place dans la mémoire, un caractère invalide détecté lors de la lecture d'un mot, un opérande manquant en essayant de dépiler un nombre pour faire une opération, une variable qui n'a pas été initialisé ou s'il reste plus d'un nombre dans la pile à la fin des calculs, le programme libère le nombre sur lequel il travaillait, s'il y en a un, enregistre un message d'erreur dans la variable globale et utilise un goto pour aller directement au bloc de traitement des erreurs. Si l'exécution du programme se trouve dans une fonction, le programme enregistre le message d'erreur et retourne un pointeur nul. Le bloc de traitement des erreurs imprime le message d'erreur, vide ce qui reste sur la pile, vide le buffer d'entrée et vide le tableau de variable temporaire et la pile de variables à libérer afin de réinitialiser les variables à leurs états d'avant la ligne erronée. Le programme passe ensuite à la prochaine ligne de calcul.