6/11/2015

Αναφορά Εργαστηρίου 2

Ομάδα

Κολομβάκης Χρήστος (Α.Μ.: 2013030103)
Ζαχαριουδάκης Χρήστος (Α.Μ.: 2014030056)

Προεργασία

Κατά την προεργασία του εργαστηρίου μας ζητήθηκε η υλοποίηση ενός προγράμματος σε γλώσσα προγραμματισμού C που θα εκτελεί τις ακόλουθες λειτουργίες :

1) Δημιουργία Λίστας
2) Εισαγωγή στοιχείου στο τέλος της λίστας
3) Διαγραφή του πρώτου στοιχείου της λίστας
4) Εκτυπώνει συγκεκριμένο στοιχείο της λίστας
5) Εκτυπώνει τον αριθμό στοιχείων της λίστας
6) Εκτύπωση διεύθυνσης συγκεκριμένου στοιχείου
7) Εκτύπωση διεύθυνσης της λίστας
8) Εκτύπωση διεύθυνσης συγκεκριμένου πεδίου για συγκεκριμένο στοιχείο
9) Εκτύπωση μεγέθους λίστας σε byte
10) Εκτύπωση μεγέθους στοιχείου σε byte
11) Έξοδος

Ομοίως στην γλώσσα προγραμματισμού CLANG , μας ζητήθηκαν οι ακόλουθες λειτουργίες :

1) Δημιουργία Λίστας
2) Εισαγωγή στοιχείου στο τέλος της λίστας
3) Διαγραφή του πρώτου στοιχείου της λίστας
4) Εκτυπώνει συγκεκριμένο στοιχείο της λίστας
5) Έξοδος

Κάθε λειτουργία εκτελείται από μια ξεχωριστή συνάρτηση και επιλέγεται από τον χρήστη μέσω ενός menu , υλοποιημένο στην main() του κώδικα μας . Ο κώδικας που υλοποιήσαμε παρατίθεται στη συνέχεια .

Περιγραφή Ζητούμενων

Ο σκοπός του 2ου εργαστηρίου ήταν η περαιτέρω εξοικείωση με την γλώσσα προγραμματισμού C και με την χρήση της λίστας ως τρόπου αποθήκευσης δεδομένων στην μνήμη. Επιπλέον , είχε σκοπό την εξάσκηση στην γλώσσα προγραμματισμού CLANG , η οποία αποτελεί μια γλώσσα με λειτουργικότητα ενδιάμεση της γλώσσας υψηλού επιπέδου C και της γλώσσας χαμηλού επιπέδου Assembly , με την οποία θα ασχοληθούμε σε επόμενα εργαστήρια .

Περιγραφής της Εκτέλεσης

Η άσκηση έγινε με χρήση του περιβάλλοντος Netbeans. Συγκεκριμένα δημιουργήσαμε 2 νέα project της C (ένα εκ των οποίων στη CLANG), από το 'File →New Project → C/C++ → C/C++ Application - >Finish'

Έπειτα από το 'Run →Build Project' κάναμε compile το πρόγραμμα μας και από το 'Run →Run Project' , το εκτελέσαμε .

Μας ζητήθηκε να δημιουργήσουμε μια λίστα 10 κόμβων (nodes) , και μετά να εκτελέσουμε τις λειτουργίες του προγράμματος .Επίσης έγινε δοκιμή των ελέγχων σφάλματος του προγράμματος , όπως εμφάνιση μηνύματος λάθους σε περίπτωση διαγραφή κόμβου σε μία κενή λίστα ή αίτησης εκτύπωσης στοιχείου που δεν υπάρχει στην λίστα .

Παράδειγμα Εκτέλεσης :

C:

The software offers you the following options :
Please enter your choice (1-11) :

1) Create a list.
2) Input a value at the end of the list.
3) Delete the first value of the list.
4) Print a value of the list.
5) Receive the number of values the list contains.
6) Print the memory address of a node.
7) Print the memory address of the list.
8) Print the memory address of the value of the node.
9) Print the size of the list in bytes.
10) Print the size of an element in bytes.
11) Exit the software.

1
Enter the value of the element.
343

MENU

1
You have already created a list!

MENU

2
Enter the value of the element.

34

MENU

2
Enter the value of the element.
34

MENU

2
Enter the value of the element.
86

MENU

2
Enter the value of the element.
53

MENU

3

MENU

4
Enter which element of the list you want printed. (1 for the first, and so on.)
1
The searched value is not found.

MENU

4
Enter which element of the list you want printed. (1 for the first, and so on.)
2
Id: 2 value: 34

MENU

5
The list contains 4 values.

MENU
3

MENU
5
The list contains 3 values.

MENU
6
Enter which element of the list you want its address printed. (1 for the first, and so on.)
2
The searched value is not found.

MENU
7
The address of the list is 584a0

MENU
8
Enter which element of the list you want printed. (1 for the first, and so on.)
2
The searched value is not found.

MENU
9
The size of the list is 48 bytes

MENU
10
The size of the node is 16 bytes.

CLANG :

The software offers you the following options :
Please enter your choice (1-5) :

1) Create a list.
2) Input a value at the end of the list.
3) Delete the first value of the list.
4) Print a value of the list.
5) Exit the software.

1
Enter the value of the element.
2

MENU

1
You have already created a list!

MENU
2
Enter the value of the element.
3

MENU
2
Enter the value of the element.
4

MENU
2
Enter the value of the element.
5

MENU
2
Enter the value of the element.
6

MENU
3

MENU
4
Enter which element of the list you want printed. (1 for the first, and so on.)
1
The searched value is not found.

MENU
4
Enter which element of the list you want printed. (1 for the first, and so on.)
2
Id: 2 value: 3

MENU
4
Enter which element of the list you want printed. (1 for the first, and so on.)
3
Id: 3 value: 4

MENU

Συμπεράσματα

Αποκτήσαμε επιπλέον εμπειρία στην χρήση της C και της λίστας δεδομένων . Μέσω της CLANG , κατανοήσαμε καλύτερα , τις λειτουργίες που εκτελεί το σύστημα στο παρασκήνιο κατά τη σύνθεση και μεταγλώττιση γλωσσών προγραμματισμού υψηλού επιπέδου , όπως η διαδικασία δέσμευσης μνήμης και εκτέλεσης πράξεων και εντολών .

Παράρτημα - Κώδικας

A) ***Υλοποιήση σε C***

```
#include <stdio.h>
#include <stdlib.h>

struct list
{
        int id  ;
        short int value ;
        struct list * next ;
};

struct list *createList (int v, short int value)  // Create a list with a node
{
        struct list * head ;

        head = (struct list *) malloc (sizeof (struct list));
        head->id = v;
        head->value = value;
        head->next = NULL;
        if (head!= NULL) return head ;
        else printf ("Operation failed\n");
}

 struct list *insertNodeList(int v,short int value, struct list *head) {  // Input a node at the end of the list
         struct list *node, *returnValue;
          struct list *node2 = head;
         node = (struct list *)malloc(sizeof(struct list));
        node->id = v;
        node->value = value;
        node->next = NULL;
        if (head == NULL) {
                returnValue = node;
                return returnValue;
          }
         while (node2->next != NULL)
            node2 = node2->next;
        node2->next = node;
         return head;
        }
```

```
struct list * deleteFirstNode(struct list *head) // Διαγραφή κόμβου στη αρχή της λίστας 3
{

	struct list *ptr=head;
	struct list *tmp;

	if (ptr != NULL)
	{
		tmp = ptr;
		 ptr = ptr->next;
		 free(tmp);
		return ptr ;
	}
	else
	{
		 printf ("The list is empty.\n");
		 return NULL ;
	}
}

int numOfElements (struct list *l) // Return the number of elements (nodes) that the list has
{

	struct list * tmp = l ;
	int counter = 0 ;

	while (tmp!=NULL)
	{
		counter++ ;
		tmp = tmp->next ;
	}

	return counter ;

	}

void printElement(int id, struct list *ptr) // Searches a node with the specified id and prints its
id and value
{

	 struct list * tmp = ptr ;
	 while (tmp!=NULL )
	 {
		if (tmp->id==id){
		 printf("Id: %d value: %d\n", tmp->id, tmp->value);
		return;
	 }
		else tmp = tmp->next ;
	 }
```

```
                printf ("The searched value is not found.\n");

        }

        void printElementAddress(int id, struct list *ptr){ // Searches a node with the
specified id and prints its address

         struct list * tmp = ptr ;

         while (tmp!=NULL )
         {
                if (tmp->id==id){
                printf("Address of element: %x\n", tmp);
                return;
           }
                else tmp = tmp->next ;
         }
          printf ("The searched value is not found.\n") ;
        }

        void printFieldAddress(int id, struct list *ptr) // Searches a node with the specified id
and prints the address of one of its fields
        {

        struct list * tmp = ptr ;
        int choice;

         while (tmp!=NULL )
          {
                if (tmp->id==id){
                        printf("Enter which field's address you want printed. 1 for id, 2 for
value.\n");
                         scanf("%d", &choice);
                  if(choice==1)
                   {
                        printf("Address of id is: %x", &(tmp->id));
                  }
                   else if(choice==2){
                      printf("Address of value is: %x", &(tmp->value));
          }
                   else{
                     printf("Invalid choice.\n");
        }
         return;

   }
        else tmp = tmp->next ;
  }

  printf ("The searched value is not found.\n");
```

```
}

void printListAddress (struct list *head) //Prints the address of the list
{
        printf ("The address of the list is : %x.\n",head);
}

void printSizeOfElement () // Prints the size of a node 10
{
        printf ("The address of the list is : %d.\n",sizeof(struct list));
    return      ;
}

void printSizeOfList (struct list * head) // Prints the size of list in bytes
{
        printf ("The address of the list is : %d.\n",sizeof(struct list)*numOfElements(head));

}

int main (){

struct list * head = NULL ;
int choice ,  id = 1 , list_created = 0 ;
short int value ;

printf ("The software offers you the following options : \n");
printf ("Please enter your choice (1-11) : \n\n");

while (1) // It will exit only if the user chooses 11
{
        printf ("\n1) Create a list.\n")     ; // User Menu
        printf ("2) Input a value at the end of the list.\n")          ;
        printf ("3) Delete the first value of the list.\n")    ;
        printf ("4) Print a value of the list.\n")    ;
        printf ("5) Receive the number of values the list contains.\n")     ;
        printf ("6) Print the memory address of a node.\n")        ;
        printf ("7) Print the memory address of the list.\n")        ;
        printf ("8) Print the memory address of the value of the node.\n")          ;
        printf ("9) Print the size of the list in bytes.\n")    ;
        printf ("10) Print the size of an element in bytes.\n")       ;
        printf ("11) Exit the software.\n\n")       ;

        scanf ("%d", &choice);

        switch (choice)
        {
                case 1 :
                        if (!list_created) // If the list is not created
                        {
                printf("Enter the value of the element.\n");
                scanf("%hd", &value);
```

```
                list_created = 1 ; // Shows that a list is created .
                        head = createList(id,value);
                id++;
                        }
                        else printf ("You have already created a list!\n");
                        break;
                case 2 :
                        if (list_created) // If the list is created
                        {
                printf("Enter the value of the element.\n");
                scanf("%hd", &value);
                head = insertNodeList(id,value,head);
                        id++;
              }
                        else printf ("You must create a list first.\n")        ;
                        break;
                case 3 :
                        if (list_created)
                                head = deleteFirstNode (head);
                        else printf ("You must create a list first.\n")        ;
                        break;
                case 4 :
                        if (list_created)
                        {
        int temp;
printf("Enter which element of the list you want printed. (1 for the first, and so on.)\n");
scanf("%d", &temp);
printElement(temp,head);
}
else printf ("You must create a list first.\n")        ;
break;
                case 5 :
                        if (list_created)
                                printf ("The list contains %d
values.\n",numOfElements(head));
                        else printf ("You must create a list first.\n")        ;
                        break;
                case 6 :
                        if (list_created)
                        {
        int temp;
                          printf("Enter which element of the list you want its address printed.
(1 for the first, and so on.)\n");
                          scanf("%d", &temp);
                          printElementAddress(temp,head);
                        }
                        else printf ("You must create a list first.\n")        ;
                        break;
                case 7 :
                        if (list_created)
                                printf ("The address of the list is %x", head );
```

```
                        else printf ("You must create a list first.\n");
                        break;
                case 8 :
                        if (list_created)
                        {
        int temp;
                           printf("Enter which element of the list you want printed. (1 for the
first, and so on.)\n");
                           scanf("%d", &temp);
                                printFieldAddress(temp,head);
                        }
                        else printf ("You must create a list first.\n")        ;
                        break;
                case 9 :
                        if (list_created)
                                printf ("The size of the list is %d bytes",numOfElements
(head)*sizeof (struct list));
                        else printf ("You must create a list first.\n")        ;
                        break;
                case 10 :
                        if (list_created)
                                printf ("The size of the node is %d bytes.\n",sizeof (struct
list));
                        else printf ("You must create a list first.\n")        ;
              break ;
                case 11 :
                        return (0);
                default :
                        printf ("Please choose one of the following options.\n");
                        break ;
        }
}
}
```

B) Υλοποίηση σε CLANG

```
#include <stdio.h>
#include <stdlib.h>

int R0 = 0;

int R1 = 1, R2, R3 =2, R4 = 3 , R5 = 4, R6 = 5, R7, R8 = 0 , R9 = 1, R10, R11, R12, R13, R14, R15,
R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31;

struct list
{
        int id  ;
        short int value ;
        struct list * next ;
};
```

```
struct list *createList (int R9 ,int R10) // Create a list with a node
{
        struct list * head ;

        head = (struct list *) malloc (sizeof (struct list));
        head->id = R9;
        head->value = R10;
        head->next = NULL;
        if (head == NULL) goto elsecreate ;
         return head ;

        elsecreate :
        printf ("Operation failed\n");

}

 struct list *insertNodeList(int R9,int R10, struct list *head) {   // Input a node at the end of
the list
  struct list *node, *returnValue;
  struct list *node2 = head;
  node = (struct list *)malloc(sizeof(struct list));
  node->id = R9;
  node->value = R10;
  node->next = NULL;

  if (head != NULL) goto afterif ;
    returnValue = node;
    return returnValue;

    afterif :

  whileinsert :

  if  (node2->next == NULL)  goto afterloop ;
      node2 = node2->next;
      goto whileinsert;

      afterloop :

  node2->next = node;
  return head;
 }


struct list * deleteFirstNode(struct list *head)   // Delete the first node of the list
{

struct list *ptr=head;
struct list *tmp;
```

```
if (ptr == NULL) goto elsedelete ;

            tmp = ptr;
   ptr = ptr->next;
   free(tmp);
   return ptr ;

elsedelete :
 printf ("The list is empty.\n");
 return NULL ;
}

void printElement (int R9, struct list *ptr)  // Searches a node with the specified id and prints
its id and value
{

  struct list *tmp = ptr ;

whileprint :

  if (tmp == NULL ) goto afterprint_loop ;

            if (tmp->id != R9) goto elseprint_label ;

      printf("Id: %d value: %d\n", tmp->id, tmp->value);
      return;

            elseprint_label:
              tmp = tmp->next ;

              goto whileprint ;
  afterprint_loop :

  printf ("The searched value is not found.\n");

}

int main () {

struct list * head = NULL ;  // R7 : choice ,  R8 : list_created , R9 : id , R10 : value , R11 : head
(pointer)

printf ("The software offers you the following options : \n");
printf ("Please enter your choice (1-5) : \n\n");

while_label :

            printf ("\n1) Create a list.\n")      ; // User Menu
            printf ("2) Input a value at the end of the list.\n")            ;
            printf ("3) Delete the first value of the list.\n")     ;
            printf ("4) Print a value of the list.\n")     ;
```

```
printf ("5) Exit the software.\n\n")          ;

scanf ("%d", &R7);

if (R7 < R1 ) goto else_label ;  // default για το αρχικο switch
if (R7 > R6) goto else_label ;

if (R7 != R1) goto case2_label ; // case 1

        if (R8 != 0) goto else1_label;

                        printf("Enter the value of the element.\n");
                        scanf("%d", &R10);
                          R8 = 1 ;
                        head = createList(R9,R10);
                         R9++;
                        goto while_label ;

                else1_label :
                printf ("You have already created a list!\n");
                goto while_label ;

        case2_label :

if (R7 != R3) goto case3_label ; // case 2

                if (R8 == 0) goto else2_label ;

        printf("Enter the value of the element.\n");
        scanf("%d", &R10);
        head = insertNodeList(R9,R10,head);
        R9++;
        goto while_label ;

                else2_label :
                printf ("You have already created a list!\n");
                goto while_label;

        case3_label :

if (R7 != R4) goto case4_label ; // case 3

                if (R8 == 0) goto else3_label ;

                        head = deleteFirstNode (head);
                        goto while_label ;

                else3_label :
                printf ("You have already created a list!\n");
                goto while_label ;
```

```
case4_label :

if (R7 != R5) goto case5_label ; // case 4

                    if (R8 == 0) goto else4_label ;

 printf("Enter which element of the list you want printed. (1 for the first, and so
on.)\n");
                      scanf("%d", &R11);
                      printElement(R11,head);
                      goto while_label ;

                    else4_label : // case 5
                    printf ("You have already created a list!\n");
                    goto while_label ;


case5_label : return (0);

                    else_label :      // default
                    printf ("Please choose one of the following options.\n");


                    goto while_label ;
}
```

C) Διάγραμμα ροής (Flowchart)