Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Лабораторна робота №17

з дисципліни

«Програмування частина 2»

Виконав:
Студент групи АП-11
Братейко Вадим

Прийняв:
Чайковський І.Б.

Львів 2024

«Дослідження особливостей використання вказівників у мові С»

Мета роботи: ознайомитися з поняттям вказівник та особливостями його використання у процесі програмування.

Теоретичні відомості

Вказівник – це змінна, значенням якої є адреса деякого об'єкта (зазвичай іншої змінної) в пам'яті комп'ютера. Наприклад, якщо одна змінна містить адресу іншої змінної, то говорять, що перша змінна вказує (посилається) на другу

Важлива особливість мови С полягає в тому, що вказівник в ній типізовані. Це означає,що якщо змінна, скажімо, р має тип «вказівник на int», то значеннями змінної р можуть бути адреси лише змінних типу int та не можуть бути адреси змінних типу double чи інших.

Присвоювання вказівників. Вказівник можна використовувати в правій частині оператора присвоювання для присвоювання його значення іншому вказівнику. Якщо обидва вказівники мають один і той же тип, то виконується просте присвоювання, без перетворення типу.

Операції адресної арифметики для вказівників. У мові С допустимі тільки дві арифметичні операції над вказівниками: сумування і віднімання. Порівняння вказівників. Стандартом С дозволяється порівняння двох вказівникв. Наприклад, якщо оголошені два вказівники р і q, то наступний оператор є правильним:

if (p <q) printf ("p посилається на меншу адресу, ніж q \ n");

Приклад 1

```
#include <stdio.h>
int main(void){
int x = 99;
int *p1, *p2;
p1 =&x;
p2= p1;
/* друк значення x два рази */
printf("Значення по адресі p1 ip2: %d %d\n", *p1, *p2);
printf("значення вказівника p1 і p2: %p %p", p1, p2);
return 0;
}
```

```
Значення по адресі p1 ip2: 99 99
значення вказівника p1 і p2: 0x7fff210fba24 0x7fff210fba24
```

Приклад 2

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
void push(int i);
int pop(void);
int *tos,*p1,stack[SIZE];
int main(void)
{
```

```
    int value;
    tos = stack; // tos посилається на основу стеку
    p1 = stack; // ініціалізація p1
    do {
        printf("Ведіть значення:");
        scanf("%d", &value);
        if(value != 0)
            push(value);
        else
            printf("Значення на вершині рівне %d\n", pop());
    } while(value != -1);
    return 0;
}

void push(int i)
{
    p1++;
    if(p1 == (tos+SIZE)) {
        printf("Perepovnennya steka.\n");
        exit(1);
    }
    *p1 = i;
}
int pop(void)
{
    if(p1 == tos) {
        printf("Stek pyctuy.\n");
        exit(1);
    }
    p1--;
    return *(p1+1);
Ведіть значення:12
Ведіть значення:-5 }
```

Приклад 3

```
#include <stdio.h>
void main(){
  int a[5];
  int sum=0;
  for (int i=0;i<5;i++){
          scanf("%d",&a[i]);
  }
  for(int j=0;j<5;j++){
          int c = *(a+j);
          sum = sum + c;
  }
```

```
  printf("sum = %d",sum);
}
```

```
5
5
9
3
3
sum = 25
```

Відповіді на контрольні запитання

1) Дайте визначення поняття вказівник:

Вказівник - це змінна, яка містить адресу пам'яті. Вона вказує на місцезнаходження (адресу) іншої змінної в пам'яті комп'ютера.

2) Які арифметичні операції можуть виконуватись з вказівниками?

З вказівниками можна виконувати наступні арифметичні операції:

Додавання (вказівник + ціле число)

Віднімання (вказівник - ціле число)

Порівняння (порівняння вказівників)

3)Поясніть призначення функції pop() у стеку:

Функція pop() в стеку використовується для видалення (або вилучення) верхнього елемента зі стеку. Це зазвичай робиться в операції видалення зі стеку.

4) Які є методи звертання до елементу масиву?

Елемент масиву можна звертатися за його індексом. Також можна використовувати вказівники для звертання до елементів масиву.

5) Які переваги використання прийомів адресної арифметики при зверненні до елементів масиву?

Використання прийомів адресної арифметики дозволяє легко переходити між елементами масиву за допомогою вказівників, що полегшує роботу з масивами. Це зроблює код більш компактним та ефективним. Також вона дозволяє працювати з масивами безпосередньо в пам'яті, що може збільшити швидкість виконання операцій.

Висновок: на цій лабораторній роботі я ознайомився з поняттям вказівник та особливостями його використання у процесі програмування.