

Języki programowania 1

Wprowadzenie do wskaźników

Jedną z potężniejszych właściwości języka C są wskaźniki. Na wstępie do wskaźników zapoznamy się z:

- zmienne wskazujące,
- adresy pamięci,
- zasada adresacji pośredniej,
- deklaracja wskaźników,
- operator adresowy,
- operator dereferencji.

Co to jest wskaźnik?

Do tej pory odwoływaliśmy się do zmiennych bezpośrednio, to znaczy używając ich nazw – identyfikatorów. Całkiem nowym zagadnieniem są wskaźniki, czyli pośrednie odwoływanie się do obiektów.

Zamiast bezpośrednio przyporządkować zmiennej jakąś wartość, możemy zmiennymi i ich zawartością manipulować pośrednio, posługując się wskazywaniem określonych adresów w pamięci RAM.

DEFINICJA: Wskaźnik (pointer) to zmienna używana do wskazywania innej zmiennej.

Wynika z tego, że wskaźnik jest zmienną. Można przyporządkowywać mu różne wartości. Wartość, którą zawiera wskaźnik musi być adresem wskazującym lokalizację w pamięci innej zmiennej. Dlatego wskaźnik jest też nazywanym zmienną adresową.

Adres (wartość lewostronna) i zawartość (wartość prawostronna)

Pamięć komputera używana jest do przechowywania kodu binarnego programu, który składa się z instrukcji i z danych, a także do przechowywania kodu binarnego systemu operacyjnego. Każda lokalizacja w pamięci musi mieć swój niepowtarzalny adres.

Również kiedy deklarujemy jakąś zmienną w programie, wolny fragment w pamięci rezerwowany jest na przechowywanie wartości danej zmiennej, a do nazwy tej zmiennej przyporządkowywany jest adres miejsca w pamięci.

Adres przyporządkowywany zmiennej jest nazywany wartością lewostronną zmiennej. Przykładowo:

```
int x;
```

```
x=7;
```

zmienna x ma dwie wartości:

wartość lewostronna: 1000 (left value), adres (nr bajtu) w pamięci, gdzie przechowywana jest wartość zmiennej x (skrzynka pocztowa z adresem)

wartość prawostronna: 7 (right value) (list włożony do naszej skrzynki pocztowej)

Operator adresowy &

Do zwracania wartości lewostronnej służy operator &. Nazywane on jest operatorem adresowym. Przykładowo:

```
int x;
```

```
y=&x;
```

powoduje przypisanie zmiennej y adresu w pamięci zmiennej całkowitej int x.

Operator & jest zazwyczaj stosowany z operatorem *, który ma dokładnie działanie odwrotne (komplementarne, uzupełniające), tzn. pobiera wartość zmiennej wskazywanej przez wskaźnik. Przykładowo:

```
int a;
```

```
a=*y;
```

oznacza pobranie zawartości spod adresu pamięci wskazywanego przez zmienną (wskaźnik) y i przyporządkuj ją jako nową wartość zmiennej a.

Otrzymywanie lewostronnych wartości zmiennych ***lewo.c***.

Deklarowanie wskaźników

Z racji tego, że wskaźnik jest całkiem normalną zmienną w programie wymaga się, aby był on również deklarowany tak jak wszystkie zmienne. Uogólniony format deklaracji wskaźnika jest następujący:

```
typ_danych *nazwa_wskaźnika;
```

Typ danych oznacza typ, na które wskaźnik wskazuje. Nazwa może być dowolna, nadana zgodnie ze standardami. Sama nazwa wskaźnika poprzedzona jest symbolem gwiazdki. Oznacza to wówczas jednoznacznie, że dana zmienna jest wskaźnikiem.

Deklarowanie i inicjowanie wskaźników w przykładzie ***wskaznik.c***.

Operator dereferencji *

Przy deklaracji wskaźnika zawsze pojawia się symbol *. Jest on nazywany operatorem dereferencji lub popularnie w żargonie komputerowym operatorem wyłuskania (operatorem pośredniego wskazania). Połączenie * i nazwy wskaźnika powoduje odwołanie się do prawostronnej wartości zmiennej wskazywanej. Przykładowo:

```
char c; /*deklaracja zmiennej*/
```

```
char *wsk_c; /*deklaracja wskaźnika*/
```

```
wsk_c=&c; /*zainicjowanie wskaźnika*/
```

Wówczas do zmiennej `c` można odwołać się w dwojaki sposób:

- bezpośrednio, `printf("%c",c),`
- pośrednio, `printf("%c",*wsk_c)`

Podobnie, jeżeli będziemy mieli:

```
int x=54, *wsk_x; /*deklaracja zmiennej i wskaźnika, zainicjowanie zmiennej*/
wsk_x=&x; /*zainicjowanie wskaźnika*/
printf("%d",(*wsk_x+1)); /*ile zostanie wyświetlone??*/
```

Modyfikowanie zawartości zmiennej przy użyciu wskaźnika do zmiennej

Jak już wiemy, po zainicjowaniu wskaźnika adresem zmiennej, można odwoływać się do zmiennej posługując się wskaźnikiem. Za pomocą operatora dereferencji `*` można odczytać z pamięci zawartość znajdującą się pod adresem wskazywanym przez wskaźnik. Podobnie można również zapisywać nową wartość zmiennej pod danym adresem. Przykład w programie ***zapis.c***.

Wskazanie tego samego adresu przez dwa wskaźniki (wiele wskaźników)

W języku C możliwe jest wskazywanie tego samego adresu przez więcej niż jeden wskaźnik. Przykładowo dwa wskaźniki typu całkowitego mogą zostać zainicjowane w taki sposób, aby wskazywały adres tej samej zmiennej.

```
int *p1, *p2;
int zmienna;
p1=&zmienna;
p2=&zmienna;
```

Praktyczne zastosowanie w programie: ***wiele.c***.