Nazwa implementacji: Nauka języka C – wskaźniki

Autor: Piotr Fiorek

Opis implementacji:

Poznanie zasad funkcjonowania pamięci w programach oraz wskaźników jako metody bezpośredniego dostępu do niej.

Wskaźniki są specjalnym typem zmiennych. Zamiast przechowywać wartości, przechowują adresy pamięci. W ten sposób można ustawić wskaźnik, aby wskazywał na adres w pamięci, pod którym znajduje się zmienna i w ten sposób, niebezpośrednio operować na tej zmiennej. Wskaźnik tak samo jak normalna zmienna musi mieć zdefiniowany typ i musi on być taki sam jak typ zmiennej, na która będzie on wskazywał. Wskaźniki deklaruje się tak samo jak zmienne, tylko nazwę poprzedzamy gwiazdką:

```
int *wskaznik;
```

Następnie do wskaźnika wartości przypisujemy tak samo jak do normalnej zmiennej tylko, że wartość przypisywana musi być poprawnym adresem w pamięci w dodatku takim, do którego nasz program ma dostęp. Najlepiej do wskaźników przypisywać po prostu adresy zmiennych. Służy do tego operator „&”. Jeśli poprzedzimy nim nazwę zmiennej, to zamiast jej wartości otrzymamy adres w pamięci, pod którym nasza zmienna się znajduje. Przypisanie adresu zmiennej do wskaźnika robimy tak:

```
int zmienna;

int *wskaznik;

wskaznik = &zmienna;
```

Aby uzyskać wartość zmiennej, na którą wskaźnik wskazuje używamy operatora „*” Prosty program pokazujący to wszystko może wyglądać tak:

```
#include <stdio.h>

int main(void)

{

int zmienna=42, zmienna2=24;

int *wskaznik;

wskaznik = &zmienna;

printf("Zmienna ma wartosc: %d\n", zmienna);

printf("Zmienna znajduje się pod adresem: %p\n\n", &zmienna);

printf("Wskaznik wskazuje na adres: %p\n", wskaznik);

printf("Zmienna pod tym adresem ma wartosc: %d\n", *wskaznik);

printf("Wskaznik znajduje się pod adresem: %p\n\n\n", &wskaznik);

*wskaznik = 5;

printf("Zmienna ma wartosc: %d\n", zmienna);

printf("Zmienna znajduje się pod adresem: %p\n\n", &zmienna);

printf("Wskaznik wskazuje na adres: %p\n", wskaznik);
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
printf("Zmienna pod tym adresem ma wartosc: %d\n", *wskaznik);

printf("Wskaznik znajduje się pod adresem: %p\n\n\n", &wskaznik);

wskaznik = &zmienna2;

printf("Zmienna ma wartosc: %d\n", zmienna2);

printf("Zmienna znajduje się pod adresem: %p\n\n", &zmienna2);

printf("Wskaznik wskazuje na adres: %p\n", wskaznik);

printf("Zmienna pod tym adresem ma wartosc: %d\n", *wskaznik);

printf("Wskaznik znajduje się pod adresem: %p\n\n\n", &wskaznik);

return 0;

}
```

Wspomniane wcześniej zapisywanie wyników operacji funkcji do zmiennej wygląda tak:

```
#include <stdio.h>

void dodawanie(int x, int y, int *suma);

int main(void)

{

int zmienna1, zmienna2, wynik;

printf("Podaj pierwsza zmienna:");

scanf("%d", &zmienna1);

printf("Podaj druga zmienna:");

scanf("%d", &zmienna2);

dodawanie(zmienna1, zmienna2, &wynik);

printf("Wynik wynosi: %d\n", wynik);

return 0;

}

void dodawanie(int x, int y, int *suma)

{

*suma = x + y;
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
}
```

Funkcja tak jak wcześniej przyjmuje dwie zmienne, które zostaną do siebie dodane, ale zamiast zwracać wynik, zapisuje go do zmiennej. Do funkcji przekazywana jest nie cała zmienna, tylko jej wartość. Dlatego właśnie przekazujemy wartość, jaką jest adres zmiennej. Adres trafia do trzeciego parametru funkcji, który jest wskaźnikiem i dalej już używamy go jak każdego innego wskaźnika. Można również deklarować wskaźniki na struktury oraz unie. Wygląda to tak samo jak w przypadku deklarowania wskaźników na zwykłe typy danych. Jedyna różnica pojawia się w momencie, kiedy chcemy się odnieść do elementu struktury, na którą wskazuje nasz wskaźnik. Można w tym celu wyciągnąć ze wskaźnika strukturę, na który wskazuje i odnieść się do jej elementów, ale jest również prostsze i bardziej eleganckie rozwiązanie i wygląda tak:

```
#include <stdio.h>

struct cyfra

{

int x;

};

int main(void)

{

struct cyfra zmienna;

struct cyfra *wskaznik;

zmienna.x = 42;

wskaznik = &zmienna;

printf("Wartosc x: %d\n", zmienna.x);

wskaznik->x = 56;

printf("Wartosc x: %d\n", wskaznik->x);

return 0;

}
```

Jak widać, aby odnieść się do elementu struktury, poprzez wskaźnik wskazujący na zmienną typu tej struktury należy użyć operatora **„->”**. Taka strzałeczka wyciągnie zmienną bez potrzeby odnoszenia się do obiektu, na który wskaźnik wskazuje. Warto dobrze zrozumieć wskaźniki i nauczyć się na nich sprawnie operować, ponieważ są one niezwykle często używane podczas programowania w C. Zadania:

Napisz program, który używając wskaźników obliczy średnią trzech liczb.

Zadanie z rozdziału o strukturach przerobić tak, aby odnoszenie się do elementów struktury odbywało się wyłącznie poprzez wskaźniki.

Napisz program obliczający miejsca zerowe funkcji kwadratowej (program musi być podzielony na funkcje realizujące poszczególne kawałki obliczenia).

Napisz program tworzący listę łączoną z elementów podawanych przez użytkownika (wykorzystaj menu z zadań o instrukcji „switch” i strukturę z zadania o strukturach).

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.