

Wskaźniki



Piotr Zierhoffer

Institute of Computer Science
Poznań University of Technology

7 października 2012

Gdzie jest zmienna?

Wyróżniamy (na razie) dwie niezależne przestrzenie pamięci (**przestrzenie adresowe**), w których przechowywane są zmienne.

Sterta

Przestrzeń do przechowywania zmiennych **dynamicznych** (w naszym wypadku - globalnych). Programista sam zarządza żywotnością tych zmiennych!

Stos

Przestrzeń przechowująca zmienne **automatyczne**, np. zmienne lokalne funkcji. Gdy opuszczamy zakres zmiennej, jej pamięć jest automatycznie zwalniana.



Zmienne w pamięci

Address	Content	Name	Type	Value
0x90000000	0x00	anInt	int	0x000000FF (255)
0x90000001	0x00			
0x90000002	0x00			
0x90000003	0xFF	aShort	short	0xFFFF (-1)
0x90000004	0xFF			
0x90000005	0xFF	aDouble	double	0x1FFFFFFFFFFFFFFF (4.4501477170144023E-308)
0x90000006	0x1F			
0x90000007	0xFF			
0x90000008	0xFF			
0x90000009	0xFF			
0x9000000A	0xFF			
0x9000000B	0xFF			
0x9000000C	0xFF	ptrAnInt	int*	0x90000000
0x9000000D	0xFF			
0x9000000E	0x90			
0x9000000F	0x00			
0x90000010	0x00			
0x90000011	0x00			

Rys.: Zmienne w pamięci

http://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp4_PointerReference.html

Wskaźnik

Co to jest wskaźnik?

Wskaźnik

To zmienna, która zawiera adres w pamięci, pod którym znajduje się inna zmienna.



Jak z tego korzystać?

```
typ *nazwa;  
int *wskInt;  
char *wskCh;  
void fun(int *wsk);
```

Mamy do dyspozycji dwa operatory:

- operator adresu &
podaje adres obiektu w pamięci
- operator odwołania pośredniego *
wskazuje na obiekt wskazywany przez wskaźnik
(*operator wyłuskania*)

Uwaga!

Nie mylimy operatora wyłuskania z deklaracją wskaźnika!



Przykład wykorzystania

```
int x = 1;
int *ip, y = 2;

ip = &x; //ip zawiera adres zmiennej x
y = *ip; //y ma wartość zmiennej, na którą wskazuje ip
*ip = 0; //zmienna, na którą wskazuje ip, ma wartość 0
printf("%d□%d\n", x, y);

printf("%p\n", ip);
///???
```

Przetestuj!



Co jeszcze trzeba wiedzieć?

- Wskaźnik wskazuje na zmienną o danym typie
uwaga na rzutowanie
- Funkcja też może zwracać wskaźnik
w następnym semestrze
- Można zadeklarować wskaźnik na wskaźnik na wskaźnik...
`int ***p;` — i to czasem też ma sens!
- Wskaźnik ma rozmiar niezależny od typu zmiennej, na którą wskazuje
ale zależny od architektury komputera! Programy w C są nieprzenośne!



Let's play

```
int a = 3;  
int b = 4;  
int *pa = &a;
```

```
*pa = b;  
pa = &b;  
a = 5;  
b = *pa;  
*pa = *pa + 10;
```



Let's play

```
int a = 3, b = 4;  
int *pa = &a;
```

```
(*pa)++;  
pa = 6;  
*pa = b;  
b = &a;
```



Wniosek

Wskaźniki to wygodne (serio!) narzędzie, ale można sobie zrobić krzywdę!

Przykład

```
int i; scanf("%d", i);
```



Okej, więc po co nam to?

Funkcje mogą zwracać jedną wartość.

Co zrobić, gdy chcemy dostać wiele wyników?

```
void swap(int a, int b)
{
    int temp;

    temp = a;
    a = b;
    b = temp;
}

int a = 6, b = 1;
swap(a, b);
printf("%d, □%d\n", a, b);
```



Poprawne podejście

Dlaczego to nie działa?

Argumenty funkcji przekazywane są jako **wartość**.

Możemy przekazać adres zmiennej:

```
void swap(int *a, int *b)
{
    int temp;

    temp = *a;
    *a = *b;
    *b = temp;
}
```

```
int a = 6, b = 1;
swap(&a, &b);
printf("%d, %d\n", a, b);
```



Na co uważać?

Mieszanie zmiennych i wskaźników:

```
int *p, a, b;
```

```
*p = a;
```

```
p = a;
```

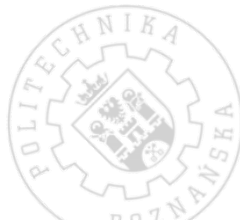
```
p = &a;
```

```
p = a;
```

```
b = p;
```

```
b = *p;
```

```
b = &p;
```



Na co uważać?

Błędy wywołań funkcji:

```
void funA(int *a);  
void funB(int b);
```

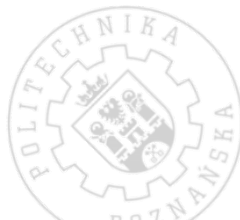
```
int *p, x;
```

```
funA(p);  
funA(*p);
```

```
funA(x);  
funA(&x);
```

```
funB(&x);  
funB(x);
```

```
funB(*p);  
funB(p);
```



Na co uważać?

Zapominanie o priorytetach operatorów:

```
*p++;  
++*p;  
(*p)++;
```

Uwaga!

Wszystkie powyższe konstrukcje mogą mieć sens w specyficznych sytuacjach.

Należy zachować ostrożność!



Zadania

Zadanie 1

Napisz funkcję, która podnosi przekazaną przez wskaźnik liczbę do kwadratu. Wczytanie liczby i wypisanie wyniku ma znaleźć się w funkcji `main`.

Zadanie 2

Napisz funkcję przyjmującą trzy parametry zmiennoprzecinkowe `a`, `b` i `c`, która sortuje ich wartości od najmniejszej do największej. Wczytanie liczb i wypisanie wyniku w funkcji `main`.



Zadania

Zadanie 3

Napisz funkcję, która wczytuje liczbę z konsoli i zapisuje ją do zmiennej przekazanej przez wskaźnik. Wywołaj ją dla trzech różnych zmiennych, wynik wypisz na ekran w funkcji `main`.



Zadania

Zadanie 4

Napisz program, który dla danej zmiennej `a` typu `short int` i wskaźnika `pa` na nią wypisze, w kolejnych liniach (z odpowiednim opisem)

- wartość zmiennej `a`
- adres zmiennej `a`
- rozmiar zmiennej `a`
- wartość zmiennej `pa`
- adres zmiennej `pa`
- rozmiar zmiennej `pa`

W komentarzach opisz każdy z otrzymanych wyników. Co one oznaczają?

