

Dyrektywy preprocesora

Wykład: preprocesor, #include, #define, #undef, #if, #endif, #else, #elseif, kompilacja warunkowa, nazwy predefiniowane, LINE, FILE, DATE, TIME

Preprocesor

to niejako przednia straż kompilatora ☺ Preprocesor przegląda kod zanim kompilator przystąpi do pracy. Bywa, iż czasem w nim coś zamieni, a czasem coś do niego doda, tak by ułatwić lub wręcz umożliwić pracę kompilatorowi. Jako wierny i oddany żołnierz, preprocesor chętnie wysłucha dodatkowych rozkazów (dyrektyw) dowódcy, czyli programisty.



`#include` - dyrektywa ta spowoduje, że w tekst kompilowanego programu zostanie wstawiona treść innego pliku z kodem - np. ciała funkcji zawartych w bibliotece

Postać dyrektywy:

```
#include <nazwa_pliku>      lub  
#include "nazwa_pliku"
```

Przykład:

```
#include <windows.h>
```

Jeżeli użyjemy znaków `<>` to pliku poszukuje się w domyślnym folderze z bibliotekami (ścieżka ustawiona w opcjach linkera), jeżeli zaś użyjemy znaków `"`, to pliku nakazujemy szukać w bieżącym folderze, a jeśli tam nie zostanie znaleziony, to będzie szukany tak jakby zamiast cudzysłowu były tam znaki `<>`.

`#define` - dyrektywa ta spowoduje, że w kompilowanym kodzie każde wystąpienie słowa `wyraz` będzie zastąpione podanym ciągiem znaków.

Postać dyrektywy:

```
#define wyraz ciąg znaków zastępujących
```

Przykład:

```
#define SIEDEM 7

cout << SIEDEM << endl;
cout << "Mowilem Ci SIEDEM razy" << endl;
cout << 10 + SIEDEM << endl;
```

`#undef` - jeżeli dyrektywą `#define` włączyliśmy zastępowanie pewnego wyrazu ciągiem znaków, to dyrektywa `#undef` wyłącza to zastępowanie.

Zatem jeżeli uprzednio w pliku nastąpiło:

```
#define wyraz ciąg znaków zastępujących
```

to po wywołaniu:

```
#undef wyraz
```

zastępowanie przestaje obowiązywać.

Przykład:

```
#undef SIEDEM
```


`#if` `#endif` `#else` `#elseif` - to dyrektywy kompilacji warunkowej. Zdarza się czasem iż chcielibyśmy, by pewne linie kodu brały udział w kompilacji jedynie jeżeli tego zażądamy - np. wersja demo, odgrywanie albo brak intro w czasie testowania gry komputerowej

Przykład:

```
#define demo 1
    cout<<"To jest kompilowane zawsze";
#if (demo==1)
    cout<<"Kompilowane tylko w demie";
#else
    cout<<"Bedzie kompilowane w pelnej wersji";
#endif
```

Nazwy predefiniowane preprocesora (występują dwa znaki podkreślenia przed i po nazwie):

`__LINE__` - kryje w sobie numer linii kodu w pliku, nad którą preprocesor aktualnie pracuje. BTW: właśnie tą informacją od preprocesora posługuje się kompilator gdy musi podać w krótej linii nastąpił błąd kompilacji)

`__FILE__` - nazwa (a czasem także ścieżka) kompilowanego właśnie pliku

`__DATE__` - data kompilacji (miesiąc dzień rok)

`__TIME__` - czas kompilacji (godzina minuta sekunda)