

2

Biblioteki i pliki nagłówkowe OpenGL-a

Aplikacja napisana w *standardzie* OpenGL musi być połączona z odpowiednią *biblioteką* OpenGL (w Linuksie `libGL.so`); biblioteki są w pakietach dystrybucji systemu operacyjnego (jeśli nie są zainstalowane domyślnie, to trzeba doinstalować odpowiedni pakiet) lub też są instalowane razem ze sterownikiem procesora graficznego dostarczonym przez jego producenta.

Oprócz biblioteki powinniśmy też mieć odpowiednie pliki nagłówkowe; podstawowy plik ma nazwę `GL/gl.h` i są w nim prototypy procedur podstawowych oraz procedur *starego* OpenGL-a, przeznaczonych do wykonywania obrazów w trybie natychmiastowym. Procedury nowego OpenGL-a są opisane w pliku `GL/glext.h`. Włączenie obu plików daje dostęp do procedur z *obu* rodzajów OpenGL-a — starego i nowego. Pisząc program, należy uważać, aby konsekwentnie używać tylko starego albo tylko nowego zestawu procedur, bo od mieszania głowa naprawdę boli. Dlatego, pisząc program *w starym stylu*, należy włączyć do aplikacji plik `GL/gl.h`¹.

Pisząc program *w nowym stylu*, zamiast pary plików `GL/gl.h` i `GL/glext.h` lepiej jest włączyć plik `GL/glcorearb.h`. Zawiera on tylko makrodefinicje, typy danych i prototypy procedur nowego OpenGL-a. Niezależnie od tego, czy używamy pary plików `GL/gl.h` i `GL/glext.h`, czy też pliku `GL/glcorearb.h`, napisanie samych dyrektyw `#include` *nie zadziała*: kompilator nie będzie „widział” prototypów procedur nowego OpenGL-a.

Aby uwidocznili prototypy procedur nowego OpenGL-a, możemy na początku programu (po innych potrzebnych dyrektywach, np. włączających pliki `stdlib.h`, `stdio.h`, `math.h` itd.) napisać

```
#define GL_GLEXT_PROTOTYPES
#include <GL/gl.h>
#include <GL/glext.h>
```

¹a po nim ewentualnie `GL/glu.h`; jest to plik nagłówkowy dodatkowej biblioteki `libGLU.so`, zawierającej wiele procedur ułatwiających pracę w trybie natychmiastowym (np. rysowanie kwadryk i powierzchni B-sklejanych). Niektóre procedury z tej biblioteki przydają się też w programach napisanych w nowym stylu.

albo

```
#define GL_GLEXT_PROTOTYPES
#include <GL/glcorearb.h>
```

Jeśli zainstalowana biblioteka OpenGL zawiera procedury wywoływane przez aplikację, to to wystarczy, aby można ją było skompilować i uruchomić. Ale skompilowana aplikacja może być przeniesiona do innego systemu, wyposażonego w bibliotekę innej wersji standardu i w szczególności zawierającej procedury realizujące inny zestaw rozszerzeń. Dlatego przygotowanie współpracy aplikacji z biblioteką OpenGL powinno się odbyć bardziej skomplikowanym sposobem.

Oto ten sposób: w katalogu z bibliotekami powinna być też biblioteka, która służy do zapewnienia współpracy aplikacji z systemem okien. Dla systemu X Window zazwyczaj dodatkowa biblioteka nie jest potrzebna, bo odpowiednie procedury są obecne w `libGL.so`². Plik nagłówkowy zawierający prototypy procedur obsługujących współpracę z systemem okien ma nazwę `GL/glx.h`. Są w nim opisane procedury, które tworzą kontekst OpenGL-a³ i przywiązują go do okna aplikacji. Oprócz tego jest tam procedura, która dla podanej nazwy dowolnej procedury OpenGL-a (łańcucha ASCIIZ) podaje adres tej procedury. Dla systemu X Window procedura podająca adresy ma nazwę `glXGetProcAddress`⁴. Podany adres może być pusty (tj. `NULL`), jeśli implementacja OpenGL-a nie zawiera procedury o takiej nazwie, bo na przykład taka procedura została określona w wersji OpenGL-a późniejszej niż wersja dostępna w danym komputerze, albo jest to procedura niestandardowa, należąca do rozszerzenia standardu dokonane przez producenta innego sprzętu niż ten, który mamy⁵.

Zatem, aplikacja napisana w sposób przenośny powinna zadeklarować odpowiednie zmienne będące wskaźnikami do procedur OpenGL-a i na początku działania „powyciągać” i przypisać tym zmiennym adresy potrzebnych procedur. Nazwy tych zmiennych wskaźnikowych powinny być identyczne z nazwami odpowiednich procedur OpenGL-a. W ten sposób na przykład instrukcja

```
glCompileShader ( shader_id );
```

ma identyczne skutki w obu przypadkach: zarówno wtedy, gdy identyfikator `glCompileShader` jest nazwą procedury dołączonej bezpośrednio do programu, jak i wtedy, gdy jest to

²Istnieje osobna biblioteka o nazwie `libGLX.so` z tymi procedurami, która może być potrzebna, jeśli podstawowa biblioteka OpenGL-a tych procedur nie zawiera, bo na przykład jest to nietypowa implementacja standardu.

³Informacje o kontekstach są w następnym rozdziale.

⁴Rozszerzeniem specyfikacji GLX 1.3 jest procedura `glXGetProcAddressARB`, w obecnej wersji 1.4 doszła `glXGetProcAddress` (w istocie *ta sama* procedura ma dwie nazwy). To jest ewidentny ślad po włączeniu rozszerzenia do następnej wersji standardu.

⁵Jeden z podstawowych mechanizmów rozwoju standardu OpenGL polega na tym, że producenci wprowadzają swoje rozszerzenia, które następnie, po podjęciu odpowiedniej decyzji przez Khronos Group, zostają włączone (albo nie) do kolejnej wersji standardu. Zresztą, rozszerzenia wprowadzone przez poszczególnych producentów, nawet jeśli nie są w standardzie, często są odtwarzane i udostępniane przez innych producentów, którzy nie chcą być gorsi.