

Biblioteki statyczne oraz pluginy

Aleksandra Poręba

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology

8 kwietnia 2019

Agenda

- » **Biblioteki statyczne**
- » Działanie bibliotek statycznych
- » Tworzenie bibliotek statycznych – Archiver
- » Użycie bibliotek statycznych
- » Wady i zalety
- » **Plugin**
- » Projektowanie - koncept
- » Narzędzia

Biblioteki statyczne

Czym są biblioteki statyczne?

- » Bibliotekami nazywamy zbiory funkcji, typów danych, etc, dostarczanych w plików, który możemy wykorzystać w naszym programie
- » Biblioteka statyczna jest zbiorem plików obiektowych, które są spakowane do jednego archiwum
- » Powstała biblioteka jest niezależna od plików, z których powstała
- » Dzięki nim unikamy konieczności każdorazowej kompilacji używanego kodu

Działanie

- » Biblioteki statyczne są łączone z programem na stałe w czasie linkowania
- » Jeśli podczas kompilacji linker nie znajdzie definicji danego symbolu, będzie jej szukał w załączonych bibliotekach
- » Odnaleziony kod zostanie dołączony
- » Każdy dołączony plik obiektowy jest czytany tylko raz

Tworzenie bibliotek statycznych

- » Jednym z narzędzi do tworzenia bibliotek statycznych jest program `Archiever`
- » Przykład 1
- » Opcje `c`, `r`, `t`, `x`
- » <http://linux.die.net/man/1/ar>
- » Tworzenie bibliotek jest odwracalne – pliki można „wypakować”
- » Przykład 2

Użycie bibliotek statycznych

» Aby użyć biblioteki statycznej należy dodać flagi:

- `-L` – ze wskazaniem położenia biblioteki
- `-l [NAZWA]` – z nazwą biblioteki

`np. gcc main.c -L./lib -lbiblioteka`

» Gdy dołączamy więcej niż jedną bibliotekę i są one od siebie zależne, należy zwrócić uwagę na kolejność dołączania

» Przykład 3

Wady i zalety

- » Program wynikowy jest przenośny - nie wymaga obecności użytych bibliotek na urządzeniu z którego korzystamy
- » Niedostępny kod źródłowy
- » Pojedynczy plik binarny zamiast wielu
- » Program wynikowy zajmuje więcej pamięci

Pluginy

Czym są pluginy

- » Pluginem, czyli inaczej wtyczką, nazywamy dodatkowe moduły do programu, które rozszerzają jego możliwości
- » Wymagają obecności programu głównego, ale on sam może działać bez ich obecności
- » Istnieją frameworki ułatwiające tworzenie wtyczek
- » Np. wtyczki w przeglądarkach (Addblock), Wireshark, Wordpress, edytory (Eclipse, IntelliJ), Valgrind

Koncepcja

- » Dodawanie lub usuwanie wtyczek nie powinno wymagać rekompilacji programu
- » Są zazwyczaj implementowane jako biblioteki dynamiczne
- » Aplikacja udostępnia **ustandaryzowany interfejs** umożliwiający wykonanie wtyczek
- » Sprowadza się to do tworzenia funkcji o zgodnych prototypach

Zastosowanie

- » Pluginy są często używane przez programistów do rozszerzania otwartego oprogramowania
- » Dzięki nim możemy łatwo dodawać nowe funkcjonalności, nie zwiększając rozmiaru aplikacji bazowej

System pluginów

- » Projektując system wspierający wtyczki należy wziąć pod uwagę następujące aspekty:
 - Poszukiwanie dostępnych pluginów
 - Rejestracja (inicjalizacja)
 - Wywoływanie funkcji
- » Najpopularniejszym schematem tworzenia wtyczek w C jest eksportowanie przez plugin wskaźników do odpowiednich funkcji, nazwanych zgodnie z przyjętym założeniem

Narzędzia

- » Oprócz frameworków (C-Pluff, Pluga, Boost.DLL) można użyć prostszych sposobów
- » Biblioteka `dlfcn.h`
- » Służy do dynamicznego linkowania
- » `dlopen()`, `dlsym()`, `dlclose()`
- » linux.pl/man/index.php?command=dlsym
- » Prosty system plugin – przykład 4

Przykład 5

- » Aplikacja kalkulatora, która posiada operacje stworzone jako pluginy
- » Interfejs dla wtyczek składa się z części poszukującej wtyczki oraz managera
- » Manager posiada dwie listy z zaczepionymi funkcjami – jeden do wypisywania opcji w menu i drugi do wykonywania działań
- » Zakładamy że każda wtyczka posiada funkcję inicjalizującą `init_[nazwa]`, która rejestruje odpowiednie funkcje w managerze

Przykład 5

- » Dwie wtyczki – `add` oraz `sub`, są tworzone jako biblioteki dynamiczne (`makefile`)
- » `plugin_discovery` – przeszukiwanie folderu z potencjalnymi wtyczkami i inicjalizowanie ich przez wywołanie funkcji `init_*`, rejestrującej funkcje
- » `plugin_manager` – struktura, która przechowuje listy ze wskaźnikami na funkcje oraz wywołuje je

Pisanie wtyczek do programów

- » Na przykładzie valgrinda
- » <http://www.valgrind.org/docs/manual/writing-tools.html>
- » Udostępniona jest instrukcja jak skonfigurować nową wtyczkę
- » Informacja o interfejsie:

A tool must define at least these four functions:

```
pre_clo_init()  
post_clo_init()  
instrument()  
fini()
```

Bibliografia

- » wikipedia.org
- » geeksforgeeks.org
- » linux.die.net/man/1/ar
- » linux.pl/man
- » eli.thegreenplace.net
- » hackaday.com/
- » Milan Stevanovic - C and C++ compiling

Dziękuję za uwagę