



PROGRAMOWANIE NISKOPOZIOMOWE

SPRAWOZDANIE
LABORATORYJNE

Binutils, biblioteki statyczne i dynamiczne

Autorzy:
Gabriel Górski
Robert Gałat

22 V 2018

1 Wstęp

Przy przygotowaniu laboratoriów staraliśmy się tworzyć zadania intensywnie wykorzystujące zagadnienia przekazane studentom w czasie naszych seminariów. Uważamy, że zadanie się udało, chociaż nie udało się uniknąć pewnych niedociągnięć.

2 Zadania

2.1 Biblioteki statyczne

- Celem zadania jest uzupełnienie pliku **run.sh** tak, aby umożliwiał on kompilację biblioteki statycznej, oraz zlinkowanie projektu do programu wynikowego. [1a]

Zalety i wady:

- + Prosto i szybko wprowadza studenta w tematykę tworzenia bibliotek statycznych
 - + Wprowadzenie do najbardziej użytecznych i fundamentalnych przy tworzeniu bibliotek flag *gcc*
 - Zadanie jest *odtwórcze* — polega jedynie na przepisaniu fragmentu konspektu do pliku skryptu
- Celem zadania jest uzupełnienie pliku **run.sh** tak, aby stworzyć biblioteki statyczne oraz zlinkować je z *głównym* plikiem obiektowym. Czy zauważasz coś ciekawego? Jeśli tak, to czy potrafisz to wyjaśnić? [1b]

Zalety i wady:

- + Zadanie jasno pokazuje znaczenie kolejności linkowania bibliotek statycznych prowokując studenta do myślenia
- Częściowy brak zrozumienia celu zadania wywołujący konsternację u studentów — konieczne było dodatkowe tłumaczenie (*Wniosek: zwiększenie stopnia interaktywności w zadaniu, przykładowo dodanie graficznej interpretacji listy brakujących referencji z kroku na krok*)

2.2 Biblioteki współdzielone

- W tym zadaniu należy utworzyć bibliotekę dynamiczną, zlinkować wobec niej plik obiektowy, a następnie otrzymany plik wykonywalny należy uruchomić — pamiętaj o odpowiednich flagach kompilacji i linkowania! [2]

Zalety i wady:

- + Łączy problematykę linkowania dynamicznego (zmienne środowiskowe) z tworzeniem i używaniem bibliotek dynamicznych
- + Utrwalenie i konfrontacja informacji zdobytych podczas poprzednich zadań - uświadomienie sobie różnic między typami bibliotek i sposobami ich tworzenia
- Nietrafiona konstrukcja konspektu oddzielająca zagadnienia bibliotek współdzielonych od zmiennych środowiskowych linkera dynamicznego wywoływała konieczność dodatkowego tłumaczenia (*Wniosek: Lepsze zgrupowanie informacji zawartych w konspekcie*)
- Celem zadania jest podmienienie implementacji funkcji która była w bibliotece z poprzedniego zadania.

Należy to zrobić bez modyfikacji pliku wykonywalnego z poprzedniego zadania tj. poprzez wykorzystanie funkcjonalności linkera dynamicznego.

Wprowadź własną implementację tej funkcji. [3]

Zalety i wady:

- + W przystępny sposób studenci starli się z dalszą obsługą linkera dynamicznego i w jaki sposób można *oszukiwać* pliki wykonywalne w odniesieniu do dostarczanych im symboli
- Brak precyzji w wymaganym nazewnictwie plików obiektowych powodujący niedziałające współzależności między zadaniami *Wniosek: Usunięcie zależności między zadaniami*

2.3 Binutils

- W tym zadaniu należy dokonać kompilacji pliku *relocatableFile.c* a następnie przeanalizować wygenerowany plik binarny programem *nm* oraz *objdump* [4]

Zalety i wady:

- + Przedstawienie fundamentalnych narzędzi do analizy plików binarnych/obiektowych
- Spreparowany przykład sugerował kompilację + linkowanie zamiast *jedynie* kompilację — konsternacja studentów
- Poświęcenie tematowi mniej czasu niż pozostałym zagadnieniom

2.4 Pluginy i dynamiczne ładowanie

- Celem zadania jest uzupełnienie pliku *main.c* w taki sposób aby uruchomić funkcję z biblioteki *libfoo.so*, która powinna zostać załadowana w czasie działania programu. [5]

Zalety i wady:

- + Przykład w doskonały i przystępny sposób prezentuje zagadnienia związane z dynamicznym ładowaniem bibliotek/symboli
- Celem zadania jest uzupełnienie brakujących części obsługi pluginów, oraz napisanie własnego pluginu, wzorując się na przygotowanym przykładzie

Do uzupełnienia są następujące funkcje:[6]

- `apply_hook()` {PluginManager/PluginManager.c}
- `initPlugin()` {PluginManager/PluginLoader.c}

Zalety i wady:

- + Angażujące zadanie dostarczające informacji o wzorcach wykorzystywanych przy budowaniu programów opartych na pluginach
- Zadanie nieco przytłaczające dla studentów, wymagające dodatkowej pomocy ze strony prowadzących

3 Ostateczne uwagi

Niestety przygotowana ilość materiału okazała się być niewystarczająca na potrzeby laboratorium. Dodatkowym problemem była znaczna różnica poziomów reprezento-

wana przez niektórych studentów. Kilka wybitnych osób skończyło 1h+ przed czasem, część — gdyby nie znaczna pomoc ze strony prowadzących — nie ukończyłaby wszystkich zadań w czasie zajęć.

Znacznym pozytywem jest jednak to, że wszystkim udało się ukończyć wszystkie zadania, mając w ten sposób możliwość przejścia przez wszystkie zawiłości i trudności opracowywanego przez nas tematu.

Wg nas wszystkie najważniejsze zagadnienia dotyczące plików obiektowych (przez najważniejsze rozumiemy wszystkie te fundamentalne, związane z przyszłą pracą inżyniera) zostały w projekcie poruszone i liczymy, że także zrozumiane przez uczestników laboratoriów oraz seminariów.