



PROGRAMOWANIE NISKOPOZIOMOWE

KONSPEKT LABORATORYJNY

---

# Binutils, biblioteki statyczne i dynamiczne

---

*Autorzy:*  
Gabriel Górski  
Robert Gałat

27 kwietnia 2018

## Spis treści

<b>1</b>	<b>Informacje do zadań</b>	<b>2</b>
<b>2</b>	<b>Zadania</b>	<b>2</b>
2.1	Biblioteki statyczne . . . . .	2
2.2	Biblioteki dynamiczne . . . . .	2
2.3	Binutils . . . . .	2
2.4	Pluginy i dynamiczne ładowanie . . . . .	2

## 1 Informacje do zadań

## 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]
- 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]

### 2.2 Biblioteki dynamiczne

- 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]
- 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]

### 2.3 Binutils

- W tym zadaniu należy dokonać kompilacji pliku *relocatableFile.c* a następnie przeanalizować wygenerowany plik binarny programem *nm* oraz *objdump* [3]

### 2.4 Pluginy i dynamiczne ładowanie

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

- 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}
- `initPlugi()` {PluginManager/PluginLoader.c}