

ОС UNIX, лекция 13: разделяемые библиотеки

Пожидаев М. С.

11 ноября 2020 г.

Использование разделяемых библиотек требует проработки трёх важных вопросов:

1. Унификация механизмов загрузки библиотек.
2. Сохранение возможности кроссплатформенной компиляции.
3. Тщательный контроль за различными моделями управления памятью.

ELF — основной формат бинарных файлов в GNU/Linux (ранее также использовался `a.out`).

`ld-linux.so` — менеджер динамически загружаемых библиотек для формата ELF. Формат `a.out` обрабатывался утилитой `ld.so`.

Переменные окружения для загрузки библиотек

- ▶ `LD_LIBRARY_PATH` — набор дополнительных каталогов для поиска динамических библиотек. Используется в дополнение к стандартным каталогам.
- ▶ `LD_PRELOAD` — набор библиотек, принудительно загружаемых вне зависимости от того, требуются они или нет.

- ▶ `ldd` — получение списка зависимостей библиотеки
- ▶ `nm` — получение списка экспортируемых функций
- ▶ `pldd` — получение списка загруженных библиотек в исполняющемся процессе

В GNU/Linux библиотека может быть загружена по явному имени файла:

- ▶ `dlopen()` — загрузка библиотеки по явному имени файла.
- ▶ `dlclose()` — выгрузка библиотеки.

```
add_library(runyms SHARED
    file1 .cpp
    file2 .cpp)

target_link_libraries (runyms lemm)
```

Примеры различий управления памятью

1. Явное освобождение (включая автоматические деструкторы).
2. Сборка мусора.
3. Что-нибудь своё (box'ы и lifetime'ы в Rust).

Пример загрузки библиотеки на Rust

```
pub fn init <'a>() -> Mystem
{
Mystem{lib: Library :: open("./libmystem_c_binding.so").expect("Error loading mystem")}
}

...

analyze: unsafe {mystem.lib.symbol_ustr::<
fn(*const u16, i32) -> u64>
(const_ustr!("MystemAnalyze").as_ustr())}
.unwrap()
```

JNI (Java Native Interface) — подсистема Java, позволяющая использовать код внешней динамически подключаемой библиотеки в программе на Java.

Порядок использования:

1. Создаётся и компилируется класс Java.
2. На его основе автоматически генерируется заголовочный файл.
3. Подготавливается код библиотеки на основе заголовочного файла.
4. Компилируется код библиотеки под каждую ОС.

Пример кода JNI

```
JNIEXPORT void JNICALL Java_org_luwrain_runyms_Runyms_initImpl
(JNIEnv* env, jobject obj)
{
    std::auto_ptr<CMorphologyHolder> h(new CMorphologyHolder());
    if (!h->LoadLemmatizer(morphRussian))
        std::cerr << "Cannot load morphology\n";
    const CMorphologyHolder* ptr = h.get();
    const jbyteArray ar = env->NewByteArray(sizeof(CMorphologyHolder*));
    env->SetByteArrayRegion(ar, 0, sizeof(CMorphologyHolder*), (jbyte*)&ptr);
    const jclass cl = env->GetObjectClass(obj);
    const jfieldID id = env->GetFieldID(cl, "core", "[B");
    env->SetObjectField(obj, id, ar);
    h.release();
}
```

Пример CMakeLists.txt для JNI

```
find_package(JNI)
if (JNI_FOUND)
    message (STATUS "JNI_INCLUDE_DIRS=${JNI_INCLUDE_DIRS}")
    message (STATUS "JNI_LIBRARIES=${JNI_LIBRARIES}")
endif ()

include_directories (${JNI_INCLUDE_DIRS})

add_library(runyms_jni SHARED jni.cpp)

target_link_libraries (runyms_jni lemm)
```

Спасибо за внимание!

Веб-сайт: <http://marigostra.ru/>

E-mail: mSP@luwrain.org