



Санкт-Петербургский государственный университет
Кафедра системного программирования

Реинжиниринг системы сборки Mybuild для Embox

Михайлов Илья Игоревич, группа 21.Б07-мм

Научный руководитель: Косарев Д. С., ассистент кафедры системного программирования

Санкт-Петербург
2024

- Системы сборки — неотъемлемая часть автоматизации разработки ПО
- Mybuild — система сборки конфигурируемой ОСРВ Embox
- Реализация Mybuild написана на GNU Make
- Предлагается написать реализацию на высокоуровневом функциональном языке OCaml

Существующие решения

- GNU Make
- Kbuild ¹
- Системы сборки использующие другие билд-системы, как бэкэнд
 - ▶ CMake, Meson ²
- Dune, Ninja

¹<https://docs.kernel.org/kbuild/index.html>

²<https://mesonbuild.com/>

Целью работы является реинжиниринг системы сборки Mybuild.

Задачи на осенний семестр:

- Реализовать парсер декларативного языка Myfile
- Протестировать парсер языка Myfile на существующей кодовой базе

Обзор Mybuild (1/4)

- Система сборки написана на синтаксическом расширении GNU Make
- Генерирует конкретный образ ОС по описаниям модулей и конфигурации
- Процесс сборки состоит из:
 - ▶ подготовки скриптов
 - ▶ создания графа описания модулей
 - ▶ создания модели системы
 - ▶ генерации необходимых ресурсов
 - ▶ запуска скриптов на исполнение

Обзор Mybuild (2/4)

В Mybuild реализовано два декларативных языка программирования:

- Myfile — для описания модулей
- Configfile — для описания конфигураций

В качестве инструмента для синтаксического анализа используется генератор LALR-парсеров GOLD ³

³<http://www.goldparser.org/>

Для каждого модуля в Build-модели необходимо:

- Включить сам модуль
- Включить все его зависимости и провести проверку (например, включить реализацию по-умолчанию для абстрактного модуля, если никакой другой реализации не было включено)
- Сопоставить значениям аргументов, указанных в конфигурационных файлах, значения опций модуля

За генерацию необходимых ресурсов отвечает скрипт `build-gen.mk`, который порождает:

- Исходные файлы на языке C, содержащие run-time представление Build-модели
- Заголовочные файлы с необходимыми опциями; также происходит копирование исходных заголовочных файлов модуля
- Make-файлы с правилами для сборки целевого образа и промежуточных объектных файлов

Реализация (1/2)

- Спроектирована структуры данных для АСТ
- Menhir был использован для синтаксического анализа
- ocamllex был использован лексического анализа

Реализация (2/2)

```
$ dune exec ../../src/parser/parse.exe /home/cy/Desktop/ocaml
  -rep/embox/src/mem/Mybuild
((Some ["embox"; "mem"]), [],
 [([] ,
  (Module
    ([], "phymem", None,
      [([], (Opt (NumberType, "log_level", (Some (
        NumberLiteral 3.)))))];
      ([], (Source ["phymem.c"]))];
      ([], (Depends [["embox"; "mem"; "vmem_device_memory"
        ]]))];
      ([], (Depends [["embox"; "mem"; "sections"]])))]))])
```

- Парсер был протестирован на файлах Mybuild из репозитория Embox, используя cram-тесты ⁴
- С помощью Github Actions была реализована непрерывная интеграция, где происходит сборка проекта при помощи dune, а также запуск тестов.

⁴https://github.com/mikhaylovilya/caravan/blob/master/test/parser_test/parse_myfiles.t
(дата обращения: 08.01.2024)

В результате проведенной работы были выполнены следующие задачи:

- 1 Написана реализация парсера декларативного языка Myfile
- 2 Проведено тестирование парсера на существующей кодовой базе в репозитории Embox

Исходный код работы доступен по ссылке ⁵. Имя аккаунта: mikhaylovilya. Название репозитория: caravan.

⁵<https://github.com/mikhaylovilya/caravan> (дата обращения: 08.01.2024)