

Сборка и непрерывная интеграция

(Часть 1. Ant)

Владимир Поляков
vladimir.p.polyakov@gmail.com

Системы сборки

Build automation software

- Ant
- Maven
- Gradle
- Leiningen
- Buildr
- ...

Зачем?

- Многоэтапная сборка сложных проектов
- Автоматизация создания дистрибутива
- Выполнение тестов
- Разрешение зависимостей
- Поддержка разных IDE

Batch

```
#!/bin/sh  
javac HelloWorld.java  
jar cvf HelloWorld.jar *  
java -jar HelloWorld.jar tests  
cp HelloWorld.jar releases/HelloWorld-v31.jar
```

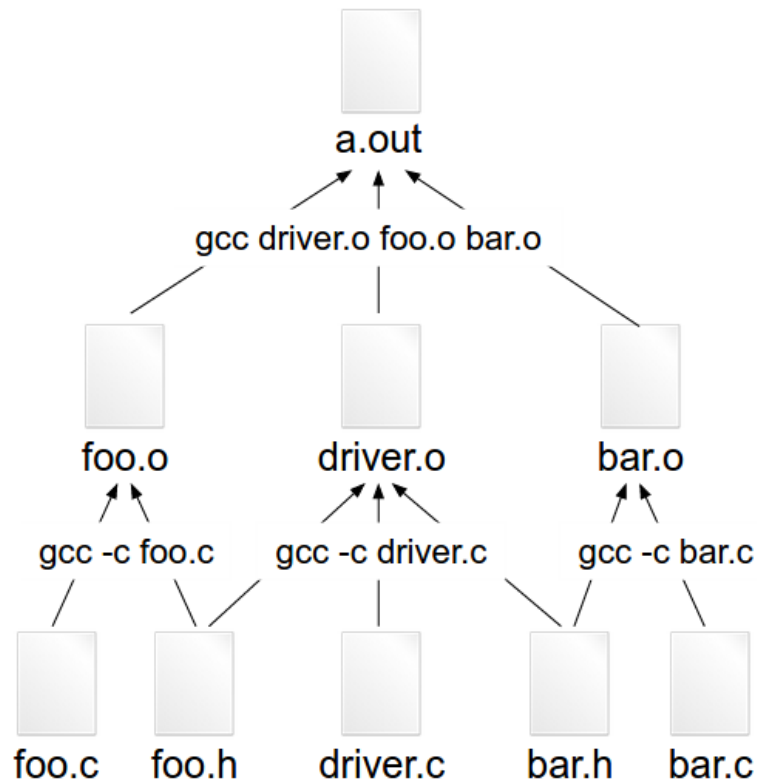
Ant

- Аналог make, но реализован на Java
- Платформонезависимый (Windows, Mac & Unix)
- Формат build-файлов - XML
- Расширяется Java-классами
- Проект с открытым исходным кодом

C & Make

```
a.out: driver.o foo.o bar.o
    gcc driver.o foo.o bar.o
driver.o: driver.c foo.h bar.h
    gcc -c driver.c
foo.o: foo.c foo.h
    gcc -c foo.c
bar.o:
    gcc -c bar.c
```

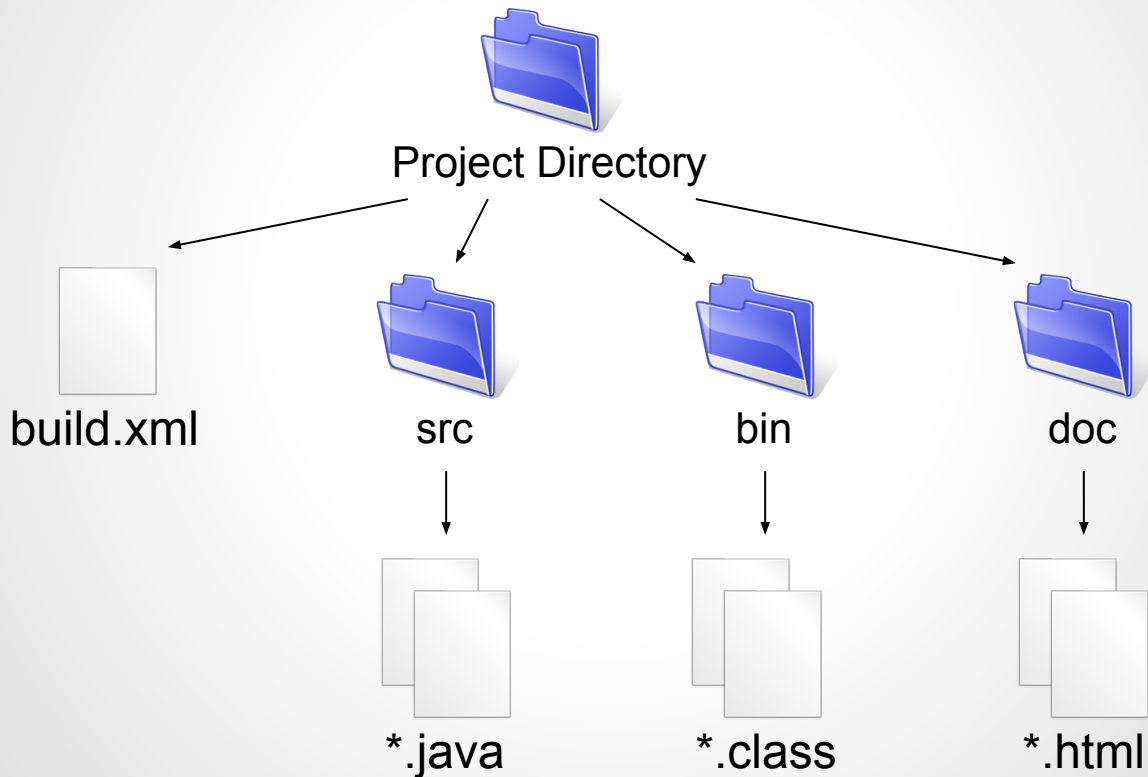
```
linux3[1]% make
gcc -c driver.c
gcc -c foo.c
gcc -c bar.c
gcc driver.o foo.o bar.o
linux3[2]%
```



Установка Ant

<http://ant.apache.org/>

Структура проекта



Build-файл

- Написан на XML
 - build.xml
- Каждый build-файл содержит
 - Проект
 - Как минимум 1 цель (target)
- Цели состоят из наборов задач

Build-файл

- Build-файл может содержать свойства (properties)
 - Как макросы в make
- И комментарии `<!-- -->`

Project

- Атрибуты
 - name – логическое имя проекта
 - default – исходная цель для выполнения
 - basedir – базовая директория
- <description> – описание проекта



Build-файл

```
<project name="Sample Project"
    default="compile"
    basedir=".">

    <description>
        A sample build file for this project
    </description>

</project>
```

Properties

- Делают проще поддержку больших build-файлов



Build-файл

```
<project name="Sample Project" default="compile" basedir=". ">
```

```
  <description>
```

```
    A sample build file for this project
```

```
  </description>
```

```
  <!-- global properties for this build file -->
```

```
  <property name="source.dir" location="src"/>
```

```
  <property name="build.dir" location="bin"/>
```

```
  <property name="doc.dir" location="doc"/>
```

```
</project>
```

Targets

- Атрибуты
 - name – имя цели
- Необязательные атрибуты
 - depends – список других целей, от которых зависит текущая
 - description – описание



Build-файл

```
<project name="Sample Project" default="compile" basedir=".">
  ...

  <!-- set up some directories used by this project -->
  <target name="init" description="setup project directories"> </target>

  <!-- Compile the java code in src dir into build dir -->
  <target name="compile" depends="init" description="compile java sources"> </target>

  <!-- Generate javadocs for current project into docs dir -->
  <target name="doc" depends="init" description="generate documentation"> </target>

  <!-- Delete the build & doc directories and Emacs backup (*~) files -->
  <target name="clean" description="tidy up the workspace"> </target>
</project>
```


Tasks

- Действие, которое нужно выполнить
- Могут иметь атрибуты, специфичные для каждой задачи
- В Ant встроен набор предопределенных задач
 - <http://ant.apache.org/manual/coretasklist.html>

Initialization target

```
<project name="Sample Project" default="compile" basedir=". ">  
  ...  
  <!-- set up some directories used by this project -->  
  <target name="init" description="setup project directories">  
    <mkdir dir="${build.dir}"/>  
    <mkdir dir="${doc.dir}"/>  
  </target>  
  ...  
</project>
```

Compilation target

```
<project name="Sample Project" default="compile" basedir=". ">
  ...
  <!-- Compile the java code in ${src.dir} into ${build.dir} -->
  <target name="compile"
    depends="init"
    description="compile java sources">
    <javac srcdir="${source.dir}" destdir="${build.dir}"/>
  </target>
  ...
</project>
```

Javadoc target

```
<project name="Sample Project" default="compile" basedir=". ">
  ...
  <!-- Generate javadocs for current project into ${doc.dir} -->
  <target name="doc"
    depends="init"
    description="generate documentation">
    <javadoc sourcepath="${source.dir}" destdir="${doc.dir}"/>
  </target>
  ...
</project>
```

Cleanup target

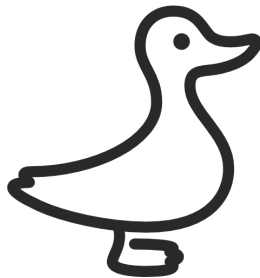
```
<project name="Sample Project" default="compile" basedir=". ">
  ...
  <!-- Delete the build & doc directories and Emacs backup (*~) files -->
  <target name="clean" description="tidy up the workspace">
    <delete dir="${build.dir}"/>
    <delete dir="${doc.dir}"/>
    <delete>
      <fileset defaultexcludes="no" dir="${source.dir}" includes="**/*~"/>
    </delete>
  </target>
  ...
</project>
```



Ant

- Коммандная строка
- Eclipse
- Idea
- ...

Вопросы



vladimir.p.polyakov@gmail.com
<https://github.com/drxaos-edu>