

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный
исследовательский университет «Московский институт электронной
техники»

Институт системной и программной инженерии и информационных
технологий (СПИНТех)

**Исследование и разработка методики и алгоритма
генерации виртуального аппаратного обеспечения
по спецификации**

Диссертация на соискание степени магистра по направлению
09.04.04 «Программная инженерия»

Научный руководитель: канд. техн. наук, доц. А. И. Кононова

Соискатель: магистрант. гр. ПИН-22М А.А. Уманский

Проблемная ситуация

Отсутствие возможности исполнения ПО разрабатываемого под специализированное аппаратное обеспечение серьезно замедляет разработку, отладку и увеличивает стоимость конечного продукта.

Причины сложившейся ситуации:

- ▶ невозможность обеспечить всех разработчиков прикладного ПО аппаратным обеспечением;
- ▶ отсутствие автоматического или полуавтоматического ПО для эмуляции аппаратного обеспечения;
- ▶ трудоемкость создания собственного виртуального аппаратного обеспечения.

Цель и задачи диссертации

Цель: снижение трудоемкости создания виртуальных устройств.

Задачи:

- ▶ аналитический обзор существующих методов создания виртуального аппаратного обеспечения;
- ▶ декомпозиция поставленной задачи для создания методики и алгоритма генерации виртуального аппаратного обеспечения;
- ▶ создание методики и алгоритма генерации виртуального аппаратного обеспечения на основе его спецификации;
- ▶ формализация задачи;
- ▶ разработка лингвистического аппарата (семантика, синтаксис) языка для создания программ по генерации виртуального аппаратного обеспечения.

Положения, выносимые на защиту

1. формализованное представление алгоритма генерации виртуального аппаратного обеспечения;
2. алгоритм генерации виртуального аппаратного обеспечения;
3. лингвистический аппарат (синтаксис, семантика) языка для создания программ по генерации виртуального аппаратного обеспечения;
4. экспериментальные результаты применения генератора аппаратного обеспечения.

Формализованное представление (грамматика)

$\langle letter \rangle ::= 'a' \dots 'z' \mid 'A' \dots 'Z';$
 $\langle digit \rangle ::= '0' \dots '9';$
 $\langle symbol \rangle ::= \backslash x20 \dots \backslash x7E; (* \text{любой печатный символ, согласно кодам ASCII} *)$
 $\langle const\ value \rangle ::= \langle digit \rangle \mid ' \{ \langle symbol \rangle \} ';$
 $\langle identifier \rangle ::= \langle letter \rangle [\{ \langle letter \rangle \mid \langle digit \rangle \mid ' _ ' \}];$
 $\langle block\ start \rangle ::= '';$
 $\langle block\ end \rangle ::= '';$
 $\langle field \rangle ::= \langle identifier \rangle '=' \langle identifier \rangle \mid \langle block \rangle;$
 $\langle block \rangle ::= \langle block\ start \rangle \langle field \rangle [\{ ' \langle field \rangle \}] \langle block\ end \rangle;$
 $\langle device\ definition \rangle ::= '\# ' \langle identifier \rangle;$
 $\langle device\ class\ inheritance \rangle ::= '(' \langle identifier \rangle ':' \langle identifier \rangle [\{ ' \langle identifier \rangle \}] ')';$
 $\langle device\ class\ block \rangle ::= \langle device\ class\ inheritance \rangle \langle block \rangle;$
 $\langle bind\ block \rangle ::= '@bind' \langle block \rangle;$
 $\langle python\ block \rangle ::= '@py' \langle block \rangle;$
 $\langle program \rangle ::= \langle device\ definition \rangle \langle device\ class\ block \rangle \langle bind\ block \rangle \langle python\ block \rangle;$

Рис.: Расширенная форма Бэкуса-Наура QPyDev

Формализованное представление (продолжение)

$$[[assignment]](x,y) = \lambda x.y$$

$[[terminate]](m)$ Завершение работы компилятора

$$[[if]](c,e_1,e_2) = \begin{cases} e_1, & \text{Если } c = true \\ e_2, & \text{Если } c \neq true \end{cases}$$

$$[[throw\ error]](c,e) = if(c,e_g,terminate)$$

$$[[lookup]](o) = [[throw\ error]](o \in Q, o)$$

$$[[< device\ definition >]](i) = lookup(i)$$

$$[[< device\ class\ inheritance >]](i_1,...,i_n) = lookup(i_1) + ... + lookup(i_n)$$

$$[[< field >]](v_1,v_2) = assignment(v_1,v_2), \text{ Если } v_1 \in Q \text{ и } v_2 \in C \cup Q$$

$$[[< block >]](f_1,...,f_n) = field(f_1) + ... + field(f_n)$$

$$[[< pythonblock >]](b) = assignment(B,B)$$

Рис.: Денотационная семантика генератора виртуального аппаратного обеспечения

Методика создания полнофункционального виртуального устройства

1. с помощью разработанного языка описываются интерфейсы и характеристики виртуального аппаратного обеспечения;
2. компилятор для разработанного языка преобразует данное описание в код на языке C;
3. скомпилированный файл с описанием виртуального устройства встраивается как новое виртуальное устройство в эмулятор QEMU;

Выбор метрики оценки эффективности

Основные метрики эффективности:

1. Время разработки виртуального устройства (в человеко-часах);
2. Быстродействие сгенерированного кода на языке C.

Основные результаты диссертационной работы

1. проведен аналитический обзор существующих методов создания виртуального аппаратного обеспечения;
2. созданы методика и алгоритм генерации виртуального аппаратного обеспечения на основе его спецификации;
3. разработан лингвистический аппарат (семантика, синтаксис) языка для создания программ по генерации виртуального аппаратного обеспечения;

Спасибо за внимание!