

Двоичная трансляция

Курс «Программное моделирование вычислительных систем»

Григорий Речистов
grigory.rechistov@phystech.edu

25 февраля 2015 г.

1 Компиляция и трансляция

2 Алгоритмы

3 Регистры, поля, банки

4 Прерывания

5 Endianness

На (поза)прошлой лекции

- Интерпретаторы — медленная шутка
- Рассмотренные улучшения основаны на повторном использовании уже полученных результатов
- Существуют устоявшиеся идиомы для представления моделируемого архитектурного состояния

Вопросы

- Определите термин «декодирование»

Вопросы

- Определите термин «декодирование»
- Сколько бит в машинном слове?

Вопросы

- Определите термин «декодирование»
- Сколько бит в машинном слове?
- Что лучше — MMIO или PIO?

Что удалось оптимизировать в интерпретаторе

- Fetch **TODO** **Напиши меня**appearify
- Decode
- Execute
- Writeback
- Advance PC

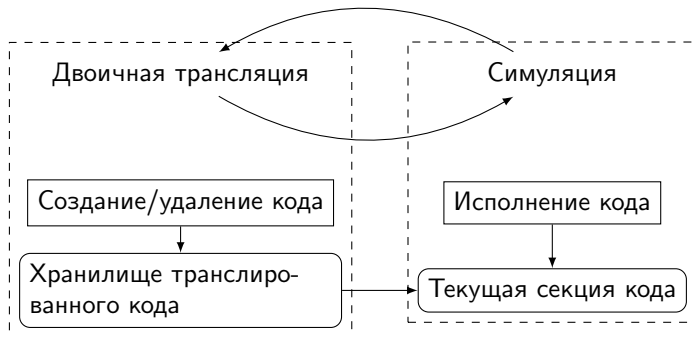
Интерпретация и трансляция в языках высокого уровня

- Basic, CPython, Shell
 - Прочитать строку - распознать команды - исполнить
 - Медленно, но больше «интерактивности»
- Fortran, C, Pascal
 - Первый проход: распознавание команд и преобразование их в машинный код
 - Второй проход: исполнение машинного кода

Двоичная трансляция

- Входной язык - гостевой машинный код
- Целевой язык - хозяйский машинный код
- ДТ - перевод кода гостевой программы, записанной в гостевой ISA, в эквивалентный код в терминах хозяйской ISA.
- Ради чего: многократное исполнение результатов трансляции
- **TODO** Напиши **меня** вопрос: что такое декомпиляция? (маш. код в ЯВО)

Фазы ДТ




Алгоритм 1 - шаблонная трансляция

```
TODO Напиши меня translate() PC = start_addr; bufptr =  
start_buf; while (! enough) instr = fetch(PC); opcode =  
decode(instr); capsule = capsules[opcode]; memcpy(capsule,  
bufptr); PC ++; bufptr ++; ; memcpy(return_jump, bufptr); ;
```

Капсула

addq (%rbx), %rax



```
push RBX_OFF(%ebp);           (1)
push (RBX_OFF+4)(%ebp);        (2)
call v2h;                      (3)
movl (%eax), %edx;             (4)
movl 4(%eax), %ebx;            (5)
addl %edx, RAX_OFF(%ebp);      (6)
addcl %ebx, 4+RAX_OFF(%ebp);   (7)
addl $3, RIP_OFF(%ebp);        (8)
```

Ленивое выделение памяти

Для непрерывного диапазона адресов хранилище выделяется по мере необходимости кусками фиксированного размера (страницами).

```
page = addr & PAGE_MASK;  
hptr = lookup_hptr(page);  
if (!hptr)  
    hptr = allocate_hptr(page);  
assert(hptr);  
haddr = hptr + (addr & PAGE_OFFSET);
```

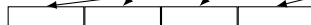
При недостатке хозяйской памяти «старые» страницы выгружаются на диск.

Звучит знакомо?

- Это же виртуальная память!
- POSIX-системы предоставляют механизм `mmap()`

```
void *mmap(void *addr,  
           size_t len, int prot, int flags,  
           int fildes, off_t off);
```

Выделенный блок



Физические страницы



Дисковый своп

Два способа взаимодействия с регистрами устройств

PIO — programmable I/O, выделенные инструкции для общения с периферией

```
IN EAX, DX  
OUT DX, EAX
```

Два способа взаимодействия с регистрами устройств

PIO — programmable I/O, выделенные инструкции для общения с периферией

```
IN EAX, DX  
OUT DX, EAX
```

MMIO — memory mapped I/O, унифицированный подход к доступу к ОЗУ и устройствам

```
MOV [mem], reg  
MOV reg, [mem]
```

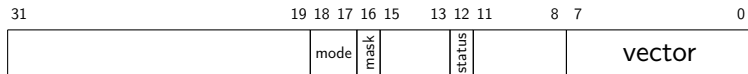

Операции над регистрами

- read, write, fetch, prefetch
- Регистры-хранилище и регистры с побочными эффектами
- Примеры регистров с side-effects: time stamp (RW), command (W), status (R), version (RO)
- inquiry — «призрачное» обращение (без эффектов)
- reset

Операции над регистрами

```
template <class rtype> class IRegister {  
    virtual Exception Read(rtype& retval) = 0;  
    virtual Exception Fetch(rtype& retval) = 0;  
    virtual Exception Prefetch(rtype& retval) = 0;  
    virtual Exception Write(const rtype& newval) = 0;  
    virtual bool InquiryRead(rtype& retval) = 0;  
    virtual bool InquiryWrite(const rtype& newval) = 0;  
    virtual void Reset() = 0;  
}
```

Битовые поля

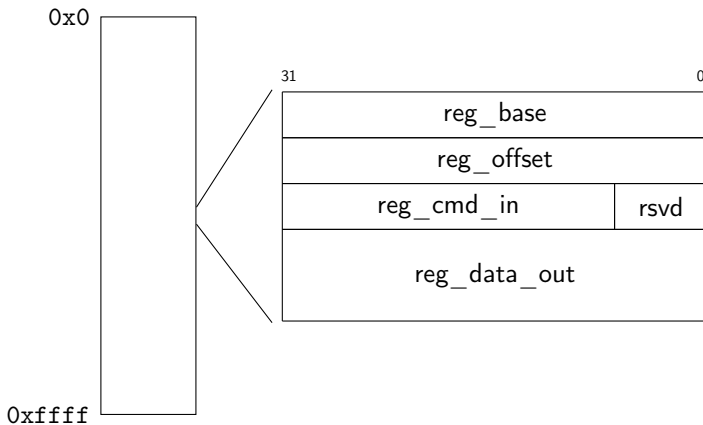


APIC LVT Timer. Intel® 64 and IA-32 Architectures Software Developer's Manual

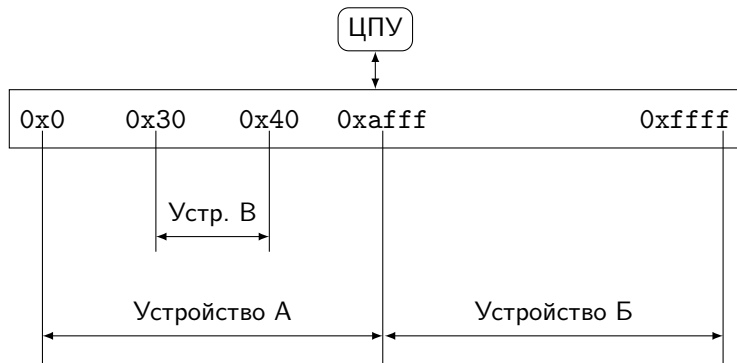
- Настоящие базовые единицы спецификаций и моделирования
- Могут иметь совершенно различные свойства внутри регистра

Банк регистров

Группа регистров устройства, находящиеся в одной области

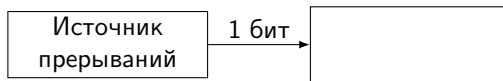


Карты памяти



Пример: devmgmt.msc

Прерывания



- Как моделировать отдельное прерывание — очень просто:
`take_interrupt(dev_t *dev);`
- Когда моделировать — вопрос сложнее, тема отдельной лекции

Преобразование адресов

- $v2p$
- $p2h$
- $v2p + p2h = v2h$

Порядок байт при доступах

Бит, байт, слово

■ Бит — ?

Бит, байт, слово

- Бит — ?
- Байт

Бит, байт, слово

- Бит — ?
- Байт — минимальная адресуемая (в данной архитектуре) единица хранения информации

Бит, байт, слово

- Бит — ?
- Байт — минимальная адресуемая (в данной архитектуре) единица хранения информации
- Октет

Бит, байт, слово

- Бит — ?
- Байт — минимальная адресуемая (в данной архитектуре) единица хранения информации
- Октет — восемь бит

Бит, байт, слово

- Бит — ?
- Байт — минимальная адресуемая (в данной архитектуре) единица хранения информации
- Октет — восемь бит
- Машинное слово

Бит, байт, слово



- Бит — ?
- Байт — минимальная адресуемая (в данной архитектуре) единица хранения информации
- Октет — восемь бит
- Машинное слово — максимальный объём информации, который ЦПУ может обработать одновременно

Бит, байт, слово

- Бит — ?
- Байт — минимальная адресуемая (в данной архитектуре) единица хранения информации
- Октет — восемь бит
- Машинное слово — максимальный объём информации, который ЦПУ может обработать одновременно

Intel: word — 16 бит, dword — 32 бит, qword — 64 бит.

Литература I

-  Stanislav Shwartsman, Darek Mihoka. How Bochs Works Under the Hood. 2nd edition. <http://bochs.sourceforge.net/HowtheBochsworksunderthehood2ndedition.pdf>
-  M. Domeika, M. Loenko, P. Ozhdikhin, E. Brevnov. Bi-Endian Compiler: A Robust and High Performance Approach for Migrating Byte Order Sensitive Applications <http://world-comp.org/p2011/ESA2902.pdf>

На следующей лекции

Ещё быстрее! Прямое исполнение



Спасибо за внимание!

Слайды и материалы курса доступны по адресу
<http://is.gd/ivuboc>

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.