

Управление ресурсами и потребность в виртуальных машинах



Николай Иготти

Ситуация - ужин в большой семье

- Одна кастрюля с супом
- Много голодных людей с разными запросами
- Как всех накормить?
- Стратегии
 - *Каждый берёт кастрюлю по очереди (сериализация монопольного доступа)*
 - *Каждому по тарелке (предварительное разделение)*
 - *Каждый подходит с тарелкой своего размера (динамическое выделение)*
 - *Каждому по виртуальной кастрюле (виртуализация, thin provisioning)*
 - *Ещё?*



Управление памятью

- Явное определение, без освобождения
- Явное выделение, явное освобождение
 - *Хорошо для небольших систем*
 - *Знакомьтесь: утечки памяти*
- Явное выделение, неявное освобождение
 - *Удобно, хотя выделять всё-таки надо*
 - *Кто же освобождает?*
 - *Когда?*
- Неявное выделение, неявное освобождение (где?)
 - *Ещё удобнее*
 - *Ещё сложнее реализовать*



Типичная ситуация

- Ограниченный ресурс (или несколько)
- Несколько потребителей с различными, часто слабопредсказуемыми, потребностями
- Стратегия индивидуального потребителя определяется его нуждами, а не наличными ресурсами
- Необходим менеджер ресурсов реализующий определённый контракт доступа
 - *malloc()/free()*
 - *new java.lang.Object()*
 - *mmap()*



Универсальный менеджер ресурсов - VM

- Рассмотрим все сущности необходимые программе для работы (контекст)
 - *Память*
 - *Процессорное время (как мы его потребляем?)*
 - *Сетевые соединения*
 - *Графические объекты*
 - *Другое*
- Ресурс – инвариант контекста
- Что если управлять всем контекстом исполнения?
- Виртуальная машина



ВМ как модель контекста

- Программы создаются в ожидании определённого контекста
- Контекст должен иметь непосредственную реализацию для исполнения программы
- Пример, контекст для:

```
#include <stdio.h>
```

```
int main(int argc, char* argv[]) {
```

```
    printf("Hello world, I'm %s\n", argv[0]);
```

```
    return 0;
```

```
}
```

ЭТО...



Что необходимо для реализации нужного контекста?

- Компьютер
- Компилятор C
- Линкер
- Загрузчик (или ОС)
- Стандартная библиотека
- Система ввода/вывода
- Виртуальная машина C?



Вопросы

1. Почему thin provisioning обычно работает?
2. Как делить неделимые ресурсы?
3. Что менеджер ресурсов должен знать о потребителях? Почему много знать плохо?
4. Где у машины Тьюринга ресурсы? Контекст?
5. Есть ли программы для которых не нужен контекст?
6. Сколько может быть реализаций одного контекста? Как (и откуда) осмысленно факторизовать реализации?

Реальная виртуальная машина ☺

Например Java Virtual Machine, содержит:

- *Загрузчик классов*
 - Получение данных класса
 - Верификация класса
- *Систему исполнения байткода*
 - Интерпретатор
 - Динамический профилировщик
 - Компилятор времени исполнения
- *Систему управления памятью*
 - Выделение пула памяти
 - Управление объектами в пуле
 - Быстрое выделение
 - Анализ доступности
 - Освобождение
- *Предзагрузчик системных классов*

Спецификация JVM

- <http://java.sun.com/docs/books/jvms/>
- Объектная модель: классы, объекты, методы
- Формат исполняемого файла
- Набор инструкций (байткод)
- Типы данных
- Абстрактная исполняющая машина
 - *Инициализация*
 - *Поток нормального исполнения*
 - *Исключительные ситуации*
 - *Реализация синхронизации потоков*



Другие необходимые части экосистемы

- Система ввода-вывода
- Интерфейс к нативному коду (JNI)
- Графический интерфейс пользователя
- Компиляторы с высокоуровневых языков (JVM != (Java) != JVM)
- Инструменты отладки
- Инструменты мониторинга и профилирования
- Среды разработки
- Среды внедрения

А зачем все эти сложности?

- Судьба индустриальных программных систем
 - *Экосистемы, а не программы*
 - *Неожиданное долголетие*
 - *Новое аппаратное обеспечение*
 - *Утеря знаний*
- Обратная совместимость, вложенная виртуализация
- Переносимость
- Управляемость
- Инерция разработчиков
- Сложность создания простого





Вопросы

1. Зачем нужен стандарт на VM?
2. Нужно ли стандартизировать что-нибудь кроме VM для Java? Почему?
3. Чего не специфицирует JVMSpec? Почему?
4. Чем плохи стандарты?
5. Где специфицирована VM языка C/C++?
6. Почему появился Jython?

Ответы, мысли: igotti@gmail.com