

# Подходы к реализации сборки мусора в ОС с персистентной памятью

Почему традиционные  
ПОДХОДЫ не применимы?

- Огромное виртуальное адресное пространство (до нескольких терабайт)

- Огромное виртуальное адресное пространство (до нескольких терабайт)
- Нет обязательной остановки программы

- Огромное виртуальное адресное пространство (до нескольких терабайт)
- Нет обязательной остановки программы
- Все приложения работают с одной общей кучей

# Предпосылки

## Phantom OS

- Снапшот виртуальной памяти

- Снапшот виртуальной памяти
- Возможно иметь полный контроль над средой (компилятор, аллокатор, сборщик мусора)м

# Мотивация

- Нельзя останавливать выполнение программ на долгое время

- Нельзя останавливать выполнение программ на долгое время
- Частичная сборка мусора

- Нельзя останавливать выполнение программ на долгое время
- Частичная сборка мусора
- Локальность объектов

- Нельзя останавливать выполнение программ на долгое время
- Частичная сборка мусора
- Локальность объектов
- Отсутствие фрагментации в долгосрочной перспективе

# Поколенческая сборка мусора

# Малая сборка мусора

- Каким должен быть размер кучи для новых объектов?

# Большая сборка мусора

- Происходит на снапшоте (tracing + compacting)

- Происходит на снапшоте (tracing + compacting)
- Из-за того, то происходит на снапшоте, она происходит асинхронно, то есть без остановки приложений

# Escape Analysis

Phantom OS

Идея

# Модификация

- Вместо стека объекты аллоцируются внутри другого специального объекта

- Вместо стека объекты аллоцируются внутри другого специального объекта
- Когда метод завершает свое выполнение, специальный объект удаляется, а вместе с ним удаляются и все временные объекты

- Вместо стека объекты аллоцируются внутри другого специального объекта
- Когда метод завершает свое выполнение, специальный объект удаляется, а вместе с ним удаляются и все временные объекты
- В лучшем случае можно очистить до 70% объектов (с медианой в 19%)

# Оптимизация

- Заменяем байт-код вызова некоторых методов на вставку всего байт-кода метода
- В пределе можно увеличить процент объектов, для которых возможно будет определить является ли объект локальным или нет

# Результаты

- Локальные объекты аллоцируются рядом друг с другом

- Локальные объекты аллоцируются рядом друг с другом
- Осуществляется освобождение части памяти за очень короткое время (практически за константу для каждого метода)

- Локальные объекты аллоцируются рядом друг с другом
- Осуществляется освобождение части памяти за очень короткое время (практически за константу для каждого метода)
- Полная сборка происходит асинхронно и в теории может быть отложена до того момента, как компьютер будет простаивать

# Вопросы?

**Зуев Антон**

a.zuev@innopolis.ru

**Дмитрий Завалишин**

dz@dz.ru