Профилирование кода

Общее описание

Чтобы профилировать приложения, компилируемые GCC, необходимо добавлять флаг `-fno -omit-frame-pointer` и, желательно, `-g`.

quickstack

Утилита для отслеживания стеков вызовов функций quicktrack. Пример использования:

```
quickstack -f -p $(pidof application)
```

perf

Утилита профилирования для ядра Linux (находится в дереве его исходных текстов в каталоге `tools/perf`).

```
perf record --call-graph dwarf -- ./application
perf report -g graph --no-children
```

Полезные ссылки:

- Официальная страница
- Примеры
- Примеры

oprofile

```
opcontrol --setup --vmlinux=/boot/vmlinux-`uname -r`
```

Systemtap

Установка и простые примеры использования SystemTap

Valgrind

Хорошая статья об использовании Valgrind для поиска утечек, а также о взаимодействии с GDB.

В версии 3.15 добавление инструмент профилирования кучи DHAT (Dynamic Heap Analysis Tool), позволяющий отследить все запросы на распределения памяти в куче и выявить утечки ресурсов, места излишне большой активности при работе с кучей, неиспользованные выделения памяти, краткосрочные выделения и неэффективное размещение данных в куче.

```
valgrind --tool=dhat ./application
```

gperftools

Разное

- Профилирование кода на C/C++ в Linux и FreeBSD
- Примеры

Графическое отображение

- Flame Graphs
- gprof2dot