

АКОС

Семинар 1

Оценка и формат работы

- Ходить можно на семинары любой группы
- В неделю за семинары нельзя получить больше 200 баллов
- Баллы за семинар выставляются за активность и решение задач
- $O_{\text{рез}} = 0.6O_{\text{накоп}} + 0.4O_{\text{экз}}$

Зачем?

- Жизненно важно для embedded, system software , security engineers:
разработка операционных систем, драйверов устройств,
низкоуровневого ПО.
- Для любого разработчика: **отладка** и **оптимизация** кода

Assembler

Наиболее близкий язык к машинному представлению команд

x86 (x86_64 / amd64)

- NASM, FASM, etc. (Intel syntax)
- GNU as (AT&T syntax)
- ...

Assembler (GAS)

- Directives (e.g. `_global`, `.text`, `.data`, `#include`, `#define`, etc.)
- Comments (`//`, `/* */`)
- Instructions
- Labels (`.my_label`)

В инструкциях в синтаксисе AT&T сначала идет source, потом destination:

```
movl $0xdeadbeef, %eax // eax = 0xdeadbeef
```

```
add %ebx, %eax // eax += ebx
```

Assembler (GAS)

- Directives (e.g. `_global`, `.text`, `.data`, `#include`, `#define`, etc.)
- Comments (`//`, `/* */`)
- Instructions (e.g. `mov`, `add`, etc.)
- Labels

В инструкциях в синтаксисе AT&T сначала идет source operand, потом destination operand:

```
movl $0xdeadbeef, %eax // eax = 0xdeadbeef
```

```
add %ebx, %eax // eax += ebx
```

Assembler (GAS)

Operands:

- Immediate value (кодируется в самой инструкции)
- Register
- *Memory*

Зачем?

Pros:

- Ручная оптимизация кода (inline assembly)
- Полный доступ к регистрам и устройствам ввода-вывода
- Возможность нарушения соглашения о вызовах и ABI в целом

Зачем?

Pros:

- **Ручная оптимизация кода**
- Полный доступ к регистрам и устройствам ввода-вывода
- Возможность нарушения соглашения о вызовах и ABI в целом

C++ source #1

A Save/Load + Add new... Vim CppInsights Quick-bench

C++

```
1 // Type your code here, or load an example.
2 unsigned square(unsigned num) {
3     return num / 3;
4 }
```

x86-64 gcc (trunk) (C++, Editor #1, Compiler #1)

x86-64 gcc (trunk)

-m32 -O0

A Output... Filter... Libraries + Add new... Add tool...

```
1 square(unsigned int):
2     push    ebp
3     mov     ebp, esp
4     mov     eax, DWORD PTR [ebp+8]
5     mov     edx, -1431655765
6     mul     edx
7     mov     eax, edx
8     shr     eax
9     pop     ebp
10    ret
```

Зачем?

Cons:

- Сложно разрабатывать и поддерживать
- Отсутствие портируемости
- Сложно масштабировать
- Error-prone

