# **AKOC**

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
- $O_{pes} = 0.6O_{HAKO\Pi} + 0.4O_{9K3}$

- 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 += abx
```

# 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 += abx
```

# Assembler (GAS)

#### Operands:

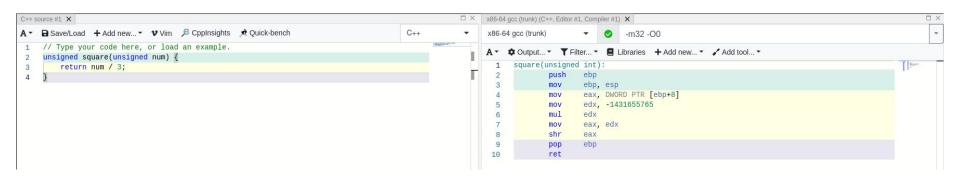
- Immediate value (кодируется в самой инструкции)
- Register
- Memory

#### Pros:

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

Pros:

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



#### Cons:

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

