

Семинар 5

AArch64

Семинар 5

! ЗАБЕЙТЕ НА ПРЕЗУ И ПОСМОТРИТЕ ЗАПИСЬ !

Prelude

- пара слов про ARM (зачем это вообще учить?)
- как собирать под ARM
- как тестировать под ARM
- как отлаживать под ARM
- напишем немного на ASM (как упростить себе жизнь)

А зачем учить что-то про ARM?

А зачем учить что-то про ARM?

Как ни странно, тут есть очень простой ответ)

А зачем учить что-то про ARM?

Как ни странно, тут есть очень простой ответ)



А зачем учить что-то про ARM?

Короче купертиновцы попытались сделать что-то и у них получилось

Маки с M1/M2 стали очень популярны, поэтому эта штука с нами на долго

А зачем учить что-то про ARM?

Короче купертиновцы попытались сделать что-то и у них получилось

Маки с M1/M2 стали очень популярны, поэтому эта штука с нами на долго

~~It's revolution Johnny~~

Как собирать под ARM

Как собирать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64

Как собирать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64
(кроме жертв пропаганды, которые решили пойти во frontend и купили маки на m1)

Как собирать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64
(кроме жертв пропаганды, которые решили пойти во frontend и купили маки на m1)
И попытка скомпилировать код обычным компилятором даст обычный x86_64 код

Как собирать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64
(кроме жертв пропаганды, которые решили пойти во frontend и купили маки на m1)
И попытка скомпилировать код обычным компилятором даст обычный x86_64 код
Следовательно нужен не обычный компилятор))

Как собирать под ARM

Для кросскомпиляции под ARM существует проект `linaro`

Вот по этой ссылке качаем

- `gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabi.tar.xz`
- `sysroot-glibc-linaro-2.25-2019.12-arm-linux-gnueabi.tar.xz`

Как собирать под ARM

<здесь делаем что-то руками>

Как тестировать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64

Как тестировать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64
(кроме жертв пропаганды, которые решили пойти во frontend и купили маки на m1)

Как тестировать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64
(кроме жертв пропаганды, которые решили пойти во frontend и купили маки на m1)

И попытка запустить ARM ELF на x86_64 платформе закончится фиаско

Как тестировать под ARM

Какие варианты?

- если у вас есть Raspberry Pi / Onion Omega 2, то пишите на них)
- для всех остальных есть Qemu

Как тестировать под ARM

Qemu

Для начала нужно поставить qemu-user (qemu user mode emulator)

```
# Ubuntu  
$ sudo apt-get install qemu-user
```

пользователи других дистрибутивов считаются продвинутыми Linux-user-ами, поэтому могут нагуглить команду самостоятельно

Как тестировать под ARM

Qemu

<тут снова делаем что-то ручками>

А как отлаживать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64

А как отлаживать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64
(кроме жертв пропаганды, которые решили пойти во frontend и купили маки на m1)

А как отлаживать под ARM

Тут вот в чем проблема... у вас у всех ноуты на x86_64
(кроме жертв пропаганды, которые решили пойти во frontend и купили маки на m1)

А чтоб отлаживать в realtime, нужно это запустить

А как отлаживать под ARM

Сори за шутки про маки на m1

Надеюсь посмеётесь над этим, когда будете сидеть в кафе, пить лавандовый раф и верстать интерфейсы...

А как отлаживать под ARM

Запускаем qemu (указываем порт для отладки)

```
qemu-arm -L ./sysroot -g 1234 ./a.out
```

Теперь читаем таблицу символов и отладочную информацию (`file`), а затем подключаемся и начинаем удаленную отладку

```
$ gdb-multiarch  
(gdb) file a.out  
(gdb) target remote localhost:1234
```

А как отлаживать под ARM

<тут снова всё ручками>