

# Задания 01

## 1. Задания

Скачайте исходный код библиотеки Unistdx из репозитория <https://github.com/igankevich/unistdx>. Эта библиотека написана для Linux и собирается системой сборки meson. Все бинарные файлы и другие артефакты сборки хранятся в отдельной директории build (название произвольное). На нашем кластере она должна быть поддиректорией /var/tmp, чтобы сборка работала корректно.

В нашем кластере 12 машин. Для того чтобы не мешать друг другу, перейдите с главной машины на другую командой `ssh мНОМЕРМАШИНЫ`, и затем выполняйте задания.

Для клонирования репозитория используйте команду

```
git clone https://github.com/igankevich/unistdx
```

1. Соберите проект с настройками по умолчанию и запустите модульные тесты.

```
meson . /var/tmp/$USER-build
cd /var/tmp/$USER-build
ninja -v
meson test
```

Здесь \$USER означает имя вашего пользователя. Флаг -v покажет команды сборки каждого файла. Запомните эти команды.

2. Соберите проект с проверкой корректности работы с памятью.

```
meson configure -Db_sanitize=address
ninja -v
meson test
```

Обратите внимание на выводимые команды сборки и появившиеся в них дополнительные флаги.

3. Соберите проект с какой-либо другой (или несколькими) проверками корректности на ваш выбор. Это делается путем указания флага -fsanitize=NNN вручную.

```
# отмена предыдущей проверки
meson configure -Db_sanitize=none
meson configure -Dcpp_args='-fsanitize=...'
ninja -v
meson test
```

Полный список возможных флагов есть в документации компилятора: `man g++`. В окне документации поиск осуществляется с помощью клавиши `/`, выход из просмотра — клавиша `q`. Обратите внимание на появившиеся в выводимых командах флаги.

4. Вычислите покрытие кода тестами.

```
# отмена предыдущей проверки
meson configure -Db_sanitize=none -Dcpp_args=''
ninja coverage-text
```

Полный список возможных команд `ninja` выводится командой `ninja -t targets`.

5. Соберите проект со статическим анализом кода.

```
ninja scan-build
```