(на сером фоне будут команды для терминала)

### I. Берём саму основную часть апертиума

1. Скачиваю по этому гайду через терминал (у меня fedora, red-hat-based, в Virtual box – отдельно могу написать как это сделать): [https://wiki.apertium.org/wiki/Install\_Apertium\_core\_using\_packaging#Red\_Hat-based](https://wiki.apertium.org/wiki/Install_Apertium_core_using_packaging" \l "Red_Hat-based)

2. Делаю sudo dnf install apertium-rus-ukr

3. Пишу echo 'Это тестовое предложение.' | apertium rus-ukr и получаю bad-alloc. Его нет, если переводить в обратную сторону.

### II. Теперь создаём новую пару языков.

https://wiki.apertium.org/wiki/How\_to\_bootstrap\_a\_new\_pair

1. для каждого языка нужно установить штуку которая отдельно про него.

а) Если язык уже есть в апертиуме: скачиваем с <https://github.com/apertium/apertium-get> файл apertium-get.py (кажется, он входит в пакет apertium-all-dev) и запускаем это с параметром в виде названия языка по коду iso-639. Пример: python3 apertium-get.py rus. В инструкции написано немножко по-другому, но это не работает.

Здесь у меня произошла некоторая ошибка, но она решилась просто тем, что я сделала ещё одну попытку с этой командой. Ошибка:

|  |
| --- |
| error: RPC failed; curl 92 HTTP/2 stream 5 was not closed cleanly: CANCEL (err 8)  error: 7080 bytes of body are still expected  fetch-pack: unexpected disconnect while reading sideband packet  fatal: неожиданный конец файла  fatal: fetch-pack: invalid index-pack output  Unable to clone apertium-rus. |

б) Если это новый язык: скачиваем скрипт apertium-init.py (<https://apertium.org/apertium-init>) и делаем это (ХХХ — код языка):

# bootstrap the module

python3 apertium-init.py XXX

(на этом моменте терминал попросил ввести ему почту и сказал, в каком формате это делать, я так и сделала: git config --global user.email “[your@e.mail](mailto:your@e.mail)” git config --global user.name “Name”)

# enter the directory

cd apertium-XXX

# compile the module

make -j

2. теперь создаём штуку про оба языка сразу.

python3 apertium-init.py XXX-YYY

cd apertium-XXX-YYY

./autogen.sh --with-lang1=../apertium-XXX --with-lang2=../apertium-YYY **(\*)**

make -j

(не пропустите, во второй строчке тоже нужны названия языков)

3. тестируем

echo дом | apertium -d . rus-olr (то есть ХХХ-YYY)

Это запускаем в папке apertium-XXX-YYY.

вывелось «\*дом», то есть он пока не знает, как перевести (да, у него нет словаря), но он понял, что ему нужно делать

Запустила то же (п. 3) на следующий день, получила ошибку Directory ‘.modes’ doesn’t exist. Решение: повторяем строчку **(\*)** и опять пишем echo.

### III. Одноязычный словарь

https://wiki.apertium.org/wiki/Building\_dictionaries

Самое понятное для начала: раздел “Одноязычные словари” здесь: https://wiki.apertium.org/wiki/Руководство\_по\_созданию\_новой\_языковой\_пары

Удобно скачать словарь для какого-нибудь уже существующего языка, и ориентироваться на него.

### IV. Двуязычный словарь

Делаю всё, что написано в разделе “Двуязычные словари” https://wiki.apertium.org/wiki/Руководство\_по\_созданию\_новой\_языковой\_пары

Но перед выполнением команд нужно поместить одноязычные словари в папку apertium-XXX-YYY с названиями типа apertium-XXX-YYY.XXX.dix и apertium-XXX-YYY.YYY.dix.