Отчет Лаба9

export GITHUB\_USERNAME= artemgarshin

export PACKAGE\_MANAGER=brew

export GPG\_PACKAGE\_NAME=gpg

$Создаем переменные

$PACKAGE\_MANAGER install xclip

$Устанавливаем утилиту для работы с буфером обмена

alias gsed=sed

alias pbcopy='xclip -selection clipboard'

alias pbpaste='xclip -selection clipboard -o'

$Переопределяем новые названия для команд

Brew install golang

$Устанавливаем утилиту golang

git clone https://github.com/${GITHUB\_USERNAME}/lab08 projects/lab09

cd projects/lab09

git remote remove origin

git remote add origin https://github.com/${GITHUB\_USERNAME}/lab09

$Клонируем репозиторий 8-ой лабораторной, переопределяем git для новой ветки нового репозитория

$PACKAGE\_MANAGER install ${GPG\_PACKAGE\_NAME}

$Устанавливаем gpg-утилиту (для создания ключей)

gpg --list-secret-keys --keyid-format LONG

gpg --full-generate-key

gpg --list-secret-keys --keyid-format LONG

gpg -K ${GITHUB\_USERNAME}

$Создаем ключи, выводим их в консоль

GPG\_KEY\_ID=$(gpg --list-secret-keys --keyid-format LONG | grep ssb | tail -1 | awk '{print $2}' | awk -F'/' '{print $2}')

GPG\_SEC\_KEY\_ID=$(gpg --list-secret-keys --keyid-format LONG | grep sec | tail -1 | awk '{print $2}' | awk -F'/' '{print $2}')

$Создаем переменные с информацией о ключах

gpg --armor --export ${GPG\_KEY\_ID} | pbcopy

$Копируем в буфер обмена ключ для вставки на GitHub

pbpaste

$Выводим его на консоль

$Далее создаем gpg ключ на github

git config user.signingkey ${GPG\_SEC\_KEY\_ID}

git config gpg.program gpg

$Добавляем информацию о ключе в репозиторий

test -r ~/.bash\_profile && echo 'export GPG\_TTY=$(tty)' >> ~/.bash\_profile

$Меняем настройки консоли

cmake -H. -B\_build -DCPACK\_GENERATOR="TGZ"

cmake --build \_build --target package

$Билдим CMake с архивом бинарных файлов

git tag -s v0.1.0.0

$Создаем защищенный тэг (флаг -s)

git tag -v v0.1.0.0

$Выводим информацию о защите тэга (флаг -v)

git show v0.1.0.0

$Выводим информацию о тэге

git push origin master --tags

$Пушим тэги

github-release --version

$Проверяем версию git-релиза

github-release info -u ${GITHUB\_USERNAME} -r lab09

$Выводим информацию о репозитории

github-release release \

--user ${GITHUB\_USERNAME} \

--repo lab09 \

--tag v0.1.0.0 \

--name "libprint" \

--description "my first release"

$Создаем релиз с параметрами (--user - логин git, --repo - название репозитория, --tag - параметр тэга, --name - название (имя) релиза, --description - описание)

export PACKAGE\_OS=`uname -s` PACKAGE\_ARCH=`uname -m`

export PACKAGE\_FILENAME=print-${PACKAGE\_OS}-${PACKAGE\_ARCH}.tar.gz

$Вводим переменные с именем нашей операционной системы, архитектурой процессора и на их основании строим переменную с именем файла

github-release upload \

--user ${GITHUB\_USERNAME} \

--repo lab09 \

--tag v0.1.0.0 \

--name "${PACKAGE\_FILENAME}" \

--file \_build/\*.tar.gz

$Заливаем в релиз файл, лежащий в build/\*.tar.gz и называем его по переменной, созданной выше

github-release info -u ${GITHUB\_USERNAME} -r lab09

$Выводим информацию о релизах репозитория

wget https://github.com/${GITHUB\_USERNAME}/lab09/releases/download/v0.1.0.0/${PACKAGE\_FILENAME}

$Скачиваем наш архив по ссылке репозитория

tar -ztf ${PACKAGE\_FILENAME}

$Разархивируем с целью проверки правильности работы