



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)
КАФЕДРА «Информационная безопасность» (ИУ8)

Лабораторная работа № 6
ПО КУРСУ
«Технологии и методы программирования»
на тему «Изучение средств пакетирования на примере
CРack»

Студент

ИУ8-24
(Группа)

К. М. Перепечкин
(И. О. Фамилия)

Преподаватель:

А. А. Кодык
(И.О. Фамилия)

2022 г.

После того, как вы настроили взаимодействие с системой непрерывной интеграции, обеспечив автоматическую сборку и тестирование ваших изменений, стоит задуматься о создании пакетов для изменений, которые помечаются тэгами (см. вкладку releases).

Пакет должен содержать приложение solver из предыдущего задания. Таким образом, каждый новый релиз будет состоять из следующих компонентов:

архивы с файлами исходного кода (.tar.gz, .zip)

пакеты с бинарным файлом solver (.deb, .rpm, .msi, .dmg)

Для этого нужно добавить ветвление в конфигурационные файлы для CI со следующей логикой: если commit помечен тэгом, то необходимо собрать пакеты (DEB, RPM, WIX, DragNDrop, ...) и разместить их на сервисе GitHub. (см. пример для Travi CI)

tets.yml:

```
1  name: packing
2  on:
3    push:
4      tags:
5        - "v*"
6
7  jobs:
8
9    pack_linux:
10     runs-on: ubuntu-latest
11     steps:
12       - name: checkout
13         uses: actions/checkout@v2
14
15       - name: build
16         shell: bash
17         run: |
18           cd solver_application/
19           cmake -B build
20           cmake --build build
21
22       - name: pack TGZ
23         shell: bash
24         run: |
25           cd solver_application/build/
26           cpack -G TGZ
27           cd ../../
28           mv solver_application/build/*.tar.gz artifacts
29
30       - name: pack ZIP
31         shell: bash
32         run: |
33           cd solver_application/build/
34           cpack -G ZIP
35           cd ../../
36           mv solver_application/build/*.zip artifacts
37
38       - name: pack DEB
39         shell: bash
40         run: |
41           cd solver_application/build/
42           cpack -G DEB
43           cd ../../
44           mv solver_application/build/*.deb artifacts
45
46       - name: pack RPM
47         shell: bash
48         run: |
49           cd solver_application/build/
50           cpack -G RPM
51           cd ../../
52           mv solver_application/build/*.rpm artifacts
53
54       - name: push on
55         shell: bash
56         run: |
57           git config --global user.name "LazuliPearl"
58           git config --global user.email "kirillperepechkin@yandex.ru"
59           cd artifacts/
60           git add *.deb *.rpm
61           git commit -m "added deb & rpm"
62           git push --force origin HEAD:master
63
64       - name: Bump version and push tag
65         id: tag_version
66         uses: mathieudutour/github-tag-action@v6.0
67         with:
```

```

70     - name: Create a GitHub release
71     uses: ncipollo/release-action@v1
72     with:
73       tag: ${ steps.tag_version.outputs.new_tag }
74       name: Release
75
76 pack_macos:
77   runs-on: macos-latest
78   steps:
79     - name: checkout
80       uses: actions/checkout@v3
81
82     - name: build
83       shell: bash
84       run: |
85         cd solver_application/
86         cmake -B build
87         cmake --build build
88
89     - name: pack DMG
90       shell: bash
91       run: |
92         cd solver_application/build/
93         cpack -G DragNDrop
94         cd ../../
95         mv solver_application/build/*.dmg artifacts
96
97     - name: push on
98       shell: bash
99       run: |
100         git config --global user.name "LazuliPearl"
101         git config --global user.email "kirillperepechkin@yandex.ru"
102         git pull origin HEAD:master
103         cd artifacts/
104         git add *.dmg
105         git commit -m "added dmg"
106         git push --force origin HEAD:master
107
108 pack_windows:
109   runs-on: windows-latest
110   steps:
111     - name: checkout
112       uses: actions/checkout@v3
113
114     - name: build
115       run: |
116         cd solver_application/
117         cmake -B build
118         cmake --build build
119
120     - name: pack MSI
121       run: |
122         cd solver_application/build/
123         cpack -G WIX
124         cd ../../
125         mv solver_application/build/*.msi artifacts
126
127     - name: push on
128       run: |
129         git config --global user.name "LazuliPearl"
130         git config --global user.email "kirillperepechkin@yandex.ru"
131         git pull origin HEAD:master
132         cd artifacts/
133         git add *.msi
134         git commit -m "added msi"

```

CPackConfig.cmake:

```
31 lines (25 sloc) | 1.12 KB
1  include(InstallRequiredSystemLibraries)
2  set(CPACK_PACKAGE_CONTACT ${GITHUB_EMAIL})
3  set(CPACK_PACKAGE_VERSION_MAJOR ${SOLVER_VERSION_MAJOR})
4  set(CPACK_PACKAGE_VERSION_MINOR ${SOLVER_VERSION_MINOR})
5  set(CPACK_PACKAGE_VERSION_PATCH ${SOLVER_VERSION_PATCH})
6  set(CPACK_PACKAGE_VERSION_TWEAK ${SOLVER_VERSION_TWEAK})
7  set(CPACK_PACKAGE_VERSION ${SOLVER_VERSION})
8  set(CPACK_PACKAGE_DESCRIPTION_FILE ${CMAKE_CURRENT_SOURCE_DIR}/DESCRIPTION)
9  set(CPACK_PACKAGE_DESCRIPTION_SUMMARY "solver app")
10
11 set(CPACK_RPM_PACKAGE_NAME "solver")
12 set(CPACK_RPM_PACKAGE_LICENSE "MIT")
13 set(CPACK_RPM_PACKAGE_GROUP "solver")
14 set(CPACK_RPM_CHANGELOG_FILE ${CMAKE_CURRENT_SOURCE_DIR}/CHANGELOG.md)
15 set(CPACK_RPM_PACKAGE_RELEASE 1)
16
17 set(CPACK_SOURCE_PACKAGE_FILE_NAME "solver-${SOLVER_VERSION}")
18 set(CPACK_SOURCE_GENERATOR "TGZ;ZIP")
19
20 set(CPACK_DEBIAN_PACKAGE_NAME "solver")
21 set(CPACK_DEBIAN_PACKAGE_DEPENDS "cmake >= 3.0")
22 set(CPACK_DEBIAN_PACKAGE_RELEASE 1)
23
24 if (CMAKE_INSTALL_PREFIX)
25     set(CPACK_OUTPUT_FILE_PREFIX ${CMAKE_INSTALL_PREFIX}/packages)
26 else()
27     set(CPACK_OUTPUT_FILE_PREFIX ${CMAKE_BINARY_DIR}/packages)
28 endif(CMAKE_INSTALL_PREFIX)
29
30 include(CPack)
31
```