

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт Информационных технологий

Кафедра корпоративных ИС

Практическая работа №1

По дисциплине: «Конфигурационное управление»

**Студент группы** ИКБО-12-22 Белоклоков Роман Валентинович

Москва, 2023

# СОДЕРЖАНИЕ

[1. Задания с доски](#40ac23ec34b54807a2c4742c6e827c20) 3

[1.1. Задание 1](#91c1cf699d90467381c423e55e0c189f) 3

[1.2. Задание 2](#57076bd5cf5b40f4816cb8ddfae22a7b) 3

[1.3. Задание 3](#3ee21425530247798ab0d64e12acd08c) 3

[1.4. Задание 4](#22d27a0d6b9d4a41820d91483a3325e4) 3

[1.5. Скриншот](#748106fba98a43f8b8156b23fb04fbe7) 3

[2. Практическое занятие №1. Введение, основы работы в командной строке](#b07083e239214a2ea769bce7149d8cef) 5

[2.1. Задание 1](#e31dbf027255455d81e4a3c7a2c8bd48) 5

[2.2. Задание 2](#f95a681087d94a5a9761c89e79e52517) 6

[2.3. Задание 3](#75bb9d455c6a4b358b95c4dacbc7d5b4) 6

[2.4. Задание 4](#dc87b00eaf964e8fa2e71ac872c936f1) 7

[2.5. Задание 5](#3b087ae5f9d94e1190947a00d968ec5d) 7

[2.6. Задание 6](#2e30fc9199854f749b0d8d924b99eda4) 8

[2.7. Задание 7](#7efd0885c9f64db69202db2fce5469e0) 8

[2.8. Задание 8](#90cf9962cde24328a71b36069ba7fd10) 8

[2.9. Задание 9](#d82f129da251415f9bf6d46bf775e25c) 9

[2.10. Задание 10](#a18459ee6dfa4658a327500d821ee6eb) 9

# Задания с доски

## Задание 1

Задание: сделать файл и папку.

● Создадим файл *file1* при помощи команды *touch*.

● Создадим папку *folder1* при помощи команды *mkdir*.

Листинг

|  |
| --- |
| touch file1  mkdir folder1 |

## Задание 2

Задание: переместить 1 в 2.

Листинг

|  |
| --- |
| mv file1 folder1 |

## Задание 3

Задание: изменить содержание файла и вывести его с дир. root.

Листинг

|  |
| --- |
| cd ..  cat folder1/file1 |

## Задание 4

Задание: написать и запустить скрипт на вывод сообщения "РТУ МИРЭА".

Листинг

|  |
| --- |
| echo "РТУ МИРЭА" |

Запустим скрипт: *sh script.sh*.

Вывод: *РТУ МИРЭА*

## Скриншот

Результат выполнения задний 1‑4 показан на рис. 1

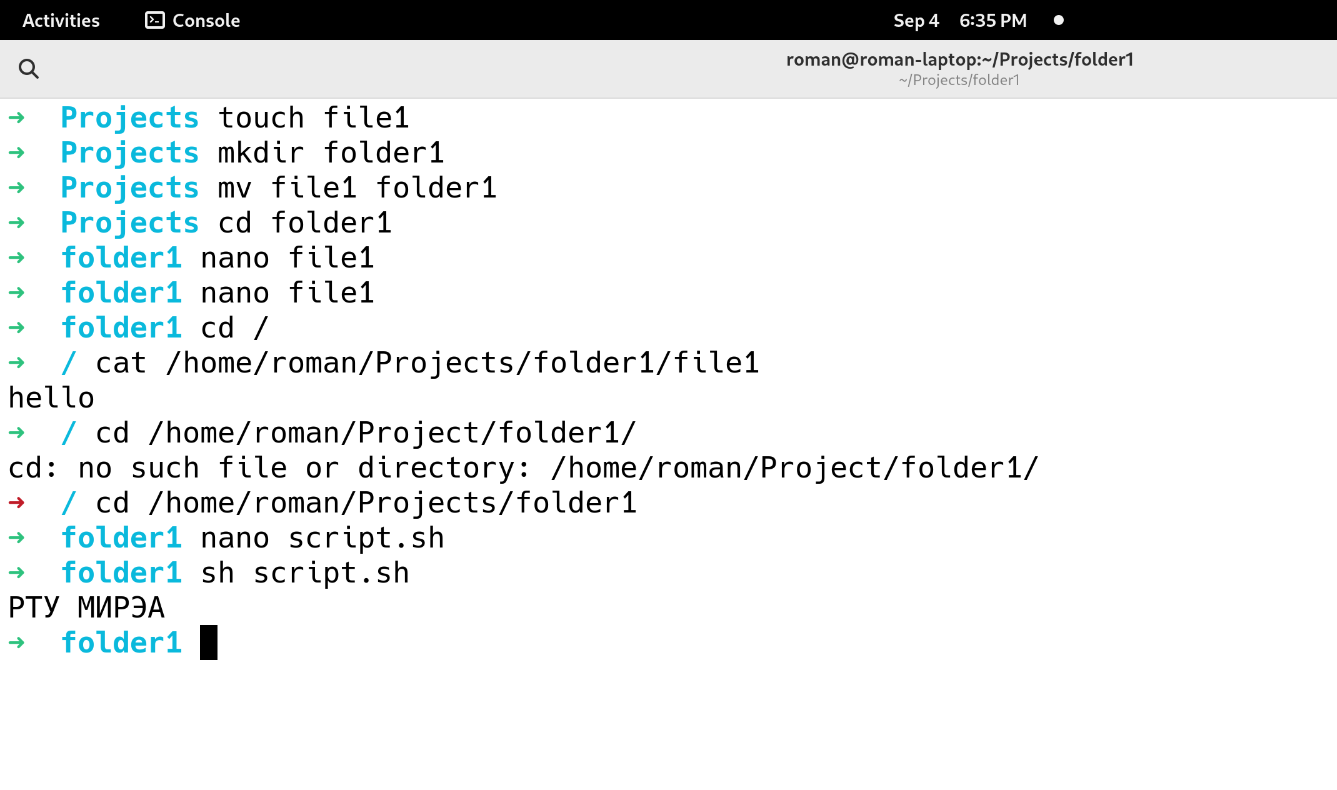


Рисунок  - Скриншот

# Практическое занятие №1. Введение, основы работы в командной строке

Выполним задания из [https://github.com/true-grue/kisscm/blob/main/pract/pract1.md‑https://github.com/true-grue/kisscm/blob/main/pract/pract1.md](https://github.com/true-grue/kisscm/blob/main/pract/pract1.md).

## Задание 1

Задание: вывести отсортированный в алфавитном порядке список имен пользователей в файле passwd (вам понадобится grep).

Выполним команду *grep* ‑*o "^[^:]\*" /etc/passwd*. Параметр ‑*o* означает что будет осуществлен вывод только найденных частей. *^* в начале паттерна означает начало строки. *[^:]\** означает любой символ кроме *:*. *\** означает, что предыдущий параметр повторяется 0 или более раз.

Вывод представлен в листинге 5.

Листинг  - Вывод пользователей из файла /etc/passwd.

|  |
| --- |
| root  bin  daemon  mail  ftp  http  nobody  dbus  systemd‑coredump  systemd‑network  systemd‑oom  systemd‑journal‑remote  systemd‑resolve  systemd‑timesync  tss  uuidd  roman  avahi  gdm  geoclue  polkitd  rtkit  usbmux  colord  git |

Продолжение листинга 5

|  |
| --- |
| nm‑openvpn  openvpn  saned  cups  plex  systemd‑journal‑upload |

## Задание 2

Задание: вывести данные /etc/protocols в отформатированном и отсортированном порядке для 5 наибольших портов.

Выполним команду *cat /etc/protocols | grep "^[^# ].+ | awk '{print $2" "$3}' | sort* ‑*n* ‑*r | head* ‑*n 5*. *cat* обозначает вывод файла. *grep* выводит только строки подходящие под паттерн: один или более любых символов, один или несколько пробелов, одна или более цифр, один или несколько пробелов, один или более любых символов. *awk* выводит вторую и третью части строки, разделенные пробелом. *sort* сортирует строки в обратном порядке (‑*r*), учитывая цифры (‑*n*). *head* выводит первые пять (‑*n 5*) строк.

Вывод представлен в листинге 6.

Листинг  - Вывод отформатированных и отсортированных данных из /etc/protcols.

|  |
| --- |
| 255 Reserved  145 NSH  144 AGGFRAG  143 Ethernet  142 ROHC |

## Задание 3

Задание: написать программу banner средствами bash для вывода текстов, как в следующем примере (размер баннера должен меняться!).

Код скрипта представлен в листинге 7. В первой строке инициализируется переменная *TEXT* со значением первого аргумента программы. Далее при помощи *expr length* подсчитывается количество символов в *TEXT*. Затем объявляется функция print\_border, которая выводит плюс, *TEXT\_LENGTH+2* минусов и плюс. Далее вызывается функция print\_boder, выводится *$TEXT*, еще раз вызывается функция print\_boder.

Листинг  - Код скрипта banner

|  |
| --- |
| **TEXT**=**$1**;  **TEXT\_LENGTH**=$(expr length "**$1**");  function print\_border {  printf "+";  for ((**i**=**1**;i<=**$TEXT\_LENGTH**+2;i++));  do  printf "‑"  done;  printf "+\n"  }  print\_border;  echo "| **$1** |";  print\_border; |

## Задание 4

Задание: Написать программу для вывода всех идентификаторов (по правилам C/C++ или Java) в файле (без повторений).

Пример для hello.c: h hello include int main n printf return stdio void world.

Листинг

|  |
| --- |
| grep ‑o ‑w ‑E "\w+" **$1** | sort ‑u |

## Задание 5

Задание: Написать программу для регистрации пользовательской команды (правильные права доступа и копирование в /usr/local/bin).

Например, пусть программа называется reg. В результате для banner задаются правильные права доступа и сам banner копируется в /usr/local/bin.

Листинг

|  |
| --- |
| chmod +x banner.sh  cp banner.sh /usr/local/bin |

## Задание 6

Задание: написать программу для проверки наличия комментария в первой строке файлов с расширением c, js и py.

Код программы предоставлен в листинге 10.

Листинг  - Код программы для задания 6.

|  |
| --- |
| **FIRST\_LINE**=$(cat **$1** | head ‑n **1**)  if { [[ **$1** == \*".py" ]] && $(echo **$FIRST\_LINE** | grep ‑q "^ \*#"); } \  || { [[ **$1** == \*".c" ]] && $(echo **$FIRST\_LINE** | grep ‑q ‑e "//.\+" ‑e "/\*.\+\\*/"); } \  || { [[ **$1** == \*".js" ]] && $(echo **$FIRST\_LINE** | grep ‑q ‑e "//.\+"); }; then  echo "comment found";  else  echo "comment not found"  fi |

## Задание 7

Задание: написать программу для нахождения файлов‑дубликатов (имеющих 1 или более копий содержимого) по заданному пути (и подкаталогам).

Листинг  - Код программы для задания 7.

|  |
| --- |
| **files**=$(find . ‑type f)  for file in **$files**; do  for file2 in **$files**; do  if [[ "**$file**" == "**$file2**" ]]; then  continue;  fi  if [[ $( cat **$file2** ) == \*$(cat **$file**)\* ]]; then  echo "duplicates **$file** **$file2**";  fi;  done  done |

## Задание 8

Задание: написать программу, которая находит все файлы в данном каталоге с расширением, указанным в качестве аргумента и архивирует все эти файлы в архив tar.

Код программы предоставлен в листинге 12.

Листинг  - Код программы для задания 8.

|  |
| --- |
| tar ‑czf archive.tar \*."**$1**" |

## Задание 9

Задание: написать программу, которая заменяет в файле последовательности из 4 пробелов на символ табуляции. Входной и выходной файлы задаются аргументами.

Код программы предоставлен в листинге 13.

Листинг  - Код программы для задания 9.

|  |
| --- |
| sed 's/ /\t/g' **$1** > **$2** |

## Задание 10

Задание: написать программу, которая выводит названия всех пустых текстовых файлов в указанной директории. Директория передается в программу параметром.

Код программы предоставлен в листинге 14.

Листинг  - Код программы для задания 10.

|  |
| --- |
| echo $(find "**$1**" ‑type f ‑empty) |