

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт Информационных технологий

Кафедра корпоративных ИС

Самостоятельная работа №1

По дисциплине: «Конфигурационное управление»

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

Москва, 2023

# Задание

Разработать эмулятор командной строки vshell. В качестве аргумента vshell принимает образ файловой системы известного формата (tar, zip).

Обратите внимание: программа должна запускаться прямо из командной строки, а файл с виртуальной файловой системой не нужно распаковывать у пользователя. В vshell должны поддерживаться команды pwd, ls, cd и cat.

Необходимо поддержать ключ командной строки ‑‑*script имя\_файла* для загрузки списка выполняемых команд из файла. Кроме того, в коде должна присутствовать функция тестирования всех реализованных команд.

Ваша задача сделать работу vshell как можно более похожей на сеанс bash в Linux. Реализовать vshell можно на Python или других ЯП, но кроссплатформенным образом.

## Исходный код

Исходный код представлен в листингах 1‑3.

Листинг  - Исходный код файла pyproject.toml

|  |
| --- |
| [tool.poetry]  name = "vshell"  version = "0.1.0"  description = ""  authors = ["Roman Beloklokov <roma.beloklockov@yandex.ru>"]  readme = "README.md"  [tool.poetry.dependencies]  python = "^3.11"  [build‑system]  requires = ["poetry‑core"]  build‑backend = "poetry.core.masonry.api"  [tool.poetry.scripts]  vshell = 'vshell.\_\_main\_\_:main' |

Листинг  - Исходный код файла vshell/\_\_main\_\_.py

|  |
| --- |
| from argparse import ArgumentParser  from .vshell import VShell |

Продолжение листинга 2

|  |
| --- |
| def **main**():  arg\_parser = ArgumentParser("vshell")  arg\_parser.add\_argument("filename", help="zip archive filename")  arg\_parser.add\_argument("‑‑script", help="путь до файла")  parsed\_args = arg\_parser.parse\_args()  vshell = VShell(parsed\_args.filename)  if parsed\_args.script:  try:  with open(parsed\_args.script) as f:  while line := f.readline():  vshell.execute\_command(line.strip())  except FileNotFoundError:  print(f"Ошибка: файл {parsed\_args.script} не существует")  else:  vshell.loop()  if **\_\_name\_\_** == "\_\_main\_\_":  main() |

Листинг  - Исходный код файла vshell/vshell.py

|  |
| --- |
| from zipfile import ZipFile, ZipInfo  from os import path, system, name as os\_name  def **split\_command**(command: str) ‑> list[str]:  res = [""]  quote = **None**  for c in command:  if c == " " and quote is **None**:  res.append("")  elif c in ("'", '"'):  if quote is **None**:  quote = c  elif quote == c:  quote = **None**  else:  res[‑**1**] += c  return res  class VShell:  def **\_\_init\_\_**(self, zip\_path: str):  self.\_zipfile = ZipFile(zip\_path)  self.\_current\_path = "/" |

Продолжение листинга 3

|  |
| --- |
| self.\_filelist = self.\_zipfile.filelist  self.\_commands = ["pwd", "ls", "cd", "cat", "exit", "clear"]  def **loop**(self):  while **True**:  self.execute\_command(input("$ "))  def **execute\_command**(self, command: str):  command = split\_command(command)  if command[**0**] in self.\_commands:  method = getattr(self, command[**0**])  else:  print("Error: Command does not exist!")  return  result = method(\*command[**1**:])  if result:  if isinstance(result, list):  print("\n".join(result))  else:  print(result)  def **parse\_path**(self, path\_: str) ‑> str:  return path.abspath(path.join(self.\_current\_path, path\_))  def **\_get\_info**(self, full\_path: str) ‑> ZipInfo | **None**:  for info in self.\_filelist:  if path.abspath("/"+info.filename) == path.abspath(full\_path):  return info  return **None**  def **pwd**(self) ‑> str:  """Prints the name of the current directory"""  return self.\_current\_path  def **ls**(self, dir: str = ".") ‑> list[str]:  """Prints the list of files in directories in the current directory"""  full\_path = self.parse\_path(dir)  if full\_path == "/":  pass  elif not (info := self.\_get\_info(full\_path)):  return "Error: Directory does not exist"  elif not info.is\_dir():  return "Error: Not a directory"  prefix = full\_path.removeprefix("/") |

Продолжение листинга 3

|  |
| --- |
| filenames = [x.filename for x in self.\_filelist]  filenames = [x.removeprefix(prefix+"/") for x in filenames  if x.startswith(prefix) and x != prefix+"/"]  return [filename for filename in filenames if "/" not in filename  or (filename.endswith("/") and filename.count("/") == **1**)]  def **cd**(self, dir: str) ‑> **None** | str:  """Changes current directory"""  full\_path = self.parse\_path(dir)  if full\_path == "/":  pass  elif not (info := self.\_get\_info(full\_path)):  return "Error: Directory does not exist"  elif not info.is\_dir():  return "Error: Not a directory"  self.\_current\_path = self.parse\_path(dir)  def **exit**(self) ‑> **None**:  """Exits the terminal"""  exit(**0**)  def **clear**(self) ‑> **None**:  """Clears the terminal screen"""  system('cls' if os\_name == 'nt' else 'clear')  def **cat**(self, filename: str) ‑> str:  """Prints file content to the terminal screen"""  full\_path = self.parse\_path(filename)  if not (info := self.\_get\_info(full\_path)):  return "Error: File does not exist"  if full\_path == "/" or info.is\_dir():  return "Error: Filename is a directory"  with self.\_zipfile.open(full\_path.removeprefix("/")) as f:  return f.read().decode("utf8") |

## Пример работы

Пример работы представлен на рис. 1.

Пример работы скрипта(листинг 4), вызванного при помощи параметра ‑‑*script* представлен на рис. 2.

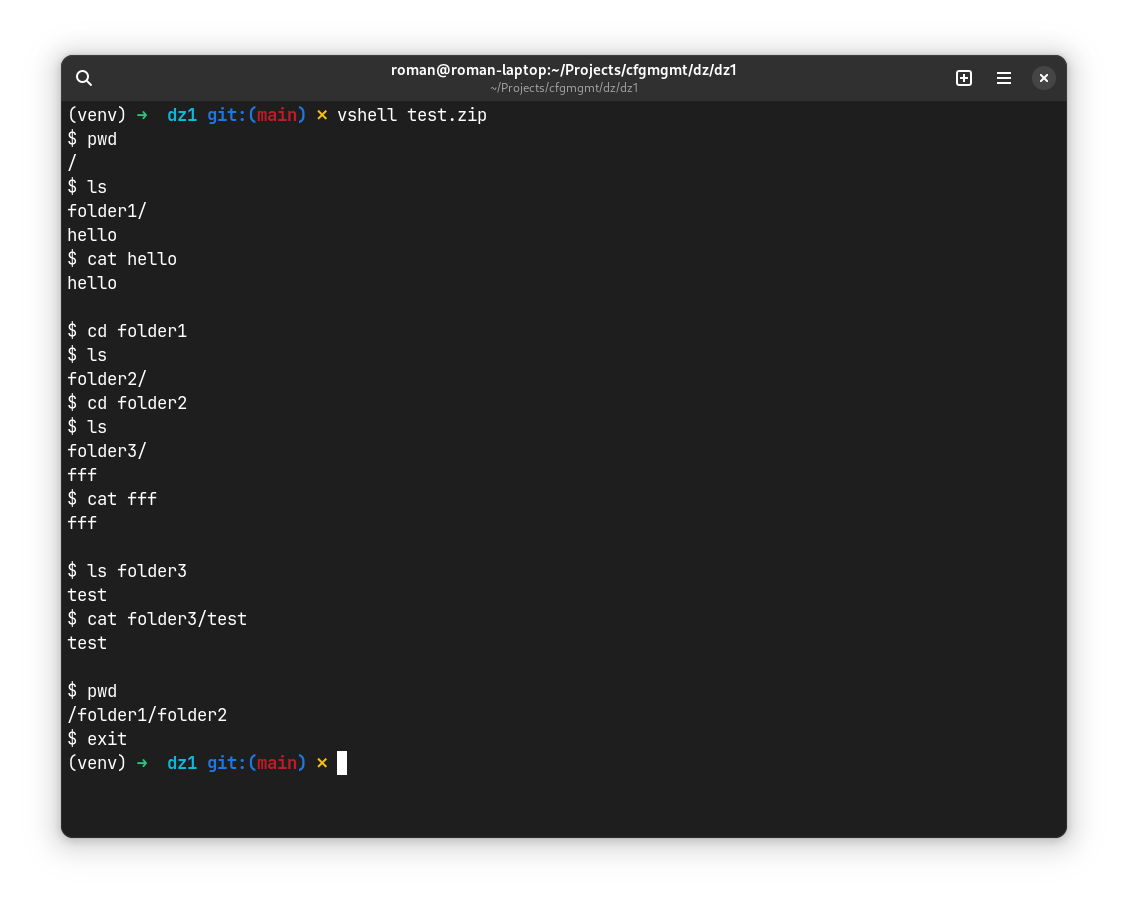


Рисунок  - Пример работы vshell

Листинг  - Код файла script.vsh

|  |
| --- |
| ls  cd folder1/folder2  cat folder3/test |

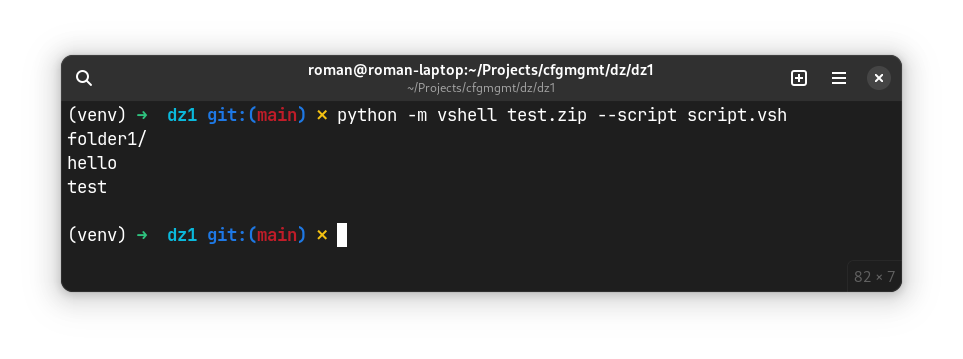


Рисунок  - Пример работы скрипта

## Вывод

При выполнения данной работы были изучены работа эмуляторов командной строки и ее простейших команд. В результате был написан свой эмулятор командной строки для zip архивов.