Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Дисциплина “Основы профессиональной деятельности”

Отчёт по лабораторной работе №1

Вариант №3274

Выполнил:

Ануфриев Андрей Сергеевич, Р3119

Проверила:

Остапенко Ольга Денисовна

г. Санкт-Петербург

2024 год

Оглавление

[Задание №1 3](#_Toc180394632)

[Задание №2 6](#_Toc180394633)

[Задание №3 7](#_Toc180394634)

[Задание №4 7](#_Toc180394635)

[Задание №5 9](#_Toc180394636)

[Ответы на вопросы 10](#_Toc180394637)

[Вывод по работе 11](#_Toc180394638)

# Задание №1

**Задание**

Создать приведенное в варианте дерево каталогов и файлов с содержимым. В качестве корня дерева использовать каталог lab0 своего домашнего каталога. Для создания и навигации по дереву использовать команды: mkdir, echo, cat, touch, ls, pwd, cd, more, cp, rm, rmdir, mv.

**Выполнение**

mkdir Lab0

cd Lab0

mkdir boldore4

cd boldore4

touch vanillish gastrodon slowking

cat > vanillish << EOF

Xoды Icy Wind Iron Defense Magic Coat Magnet Rise Signal Beam Sleep Talk Snore Uproar

EOF

cat > gastrodon << EOF

Развитые способности Sand

Force

EOF

cat > slowking << EOF

Ходы After You Aqua Tail Avalanche Block Body Slam

Brine Counter Dive Double-Edge Drain Punch Dynamicpunch Focus Punch Foul Play Fury Cutter Hidden Power+ Ice Punch Icy Wind Iron Defense Iron Tail Magic Coat Mega Kick Mega Punch Mud-Slap Power GemвЂЎ Recycle Role Play Seismic Toss Signal Beam Skill Swap Sleep Talk Snore Swift Trick Wonder Room Zen Headbutt

EOF

cd ..

mkdir crustle6

cd crustle6

touch stunky volcarona blastoise krabby

cat > stunky << EOF

Тип покемона POISON

DARK

EOF

cat > volcarona << EOF

Развитые способности Flare

Boost

EOF

cat > blastoise << EOF

Возможности Overland 4 Surface=12 Underwater=10 Jump1=0 Power=4 Intelligence=4 Fountain=0

EOF

cat > krabby << EOF

Развитые способности

Sheer Force

EOF

mkdir sealeo altaria

cd ..

mkdir hoothoot0

cd hoothoot0

touch koffing doduo gorebyss jolteon totodile

cat > koffing << EOF

Способности Venom Levitate White

Smoke

EOF

cat > doduo << EOF

weigth=86.4 height=55.0 atk=9 def=5

EOF

cat > gorebyss << EOF

Живет

Ocean

EOF

cat > jolteon << EOF

Способности Overcharge Volt Absorb Motor

Drive

EOF

cat > totodile << EOF

Живет Beach Freshwater Marsh

EOF

mkdir abomasnow

cd ..

touch grotle5 ninetales2 quagsire9

cat > grotle5 << EOF

Способности Withdraw Absorb Razor Leaf Curse

Bite Mega Drain Leech Seed Synthesis Crunch Giga Drain Leaf Storm

EOF

cat > ninetales2 << EOF

Живет

Grassland Mountain

EOF

cat > quagsire9 << EOF

Возможности Overland=7 Surface=8

Underwater=7 Jump 3 Power=3 Intelligence=4 Fountain=0 Gilled=0

EOF

cd ..

cd ..

# Задание №2

**Задание**

Установить согласно заданию права на файлы и каталоги при помощи команды chmod, используя различные способы указания прав.

**Выполнение**

cd Lab0

chmod 361 boldore4

cd boldore4

chmod u=rw,g=w vanillish

chmod u=r gastrodon

chmod g=r,o=rw slowking

cd ..

chmod 752 crustle6

cd crustle6

chmod 351 sealeo

chmod 404 stunky

chmod 620 volcarona

chmod u=r,o=r altaria

chmod 046 blastoise

chmod 062 krabby

cd ..

chmod 622 grotle5

chmod 357 hoothoot0

cd hoothoot0

chmod 440 koffing

chmod 400 doduo

chmod 444 gorebyss

chmod 737 abomasnow

chmod 400 jolteon

chmod 604 totodile

cd ..

chmod 660 ninetales2

chmod 440 quagsire9

cd ..

cd ..

# Задание №3

**Задание**

Скопировать часть дерева и создать ссылки внутри дерева согласно заданию при помощи команд cp и ln, а также комманды cat и перенаправления ввода-вывода.

**Выполнение**

cd Lab0

ln -s boldore4 Copy\_91

ln -s ninetales2 hoothoot0/totodileninetales

ln quagsire9 hoothoot0/koffingquagsire

cp -r hoothoot0/ crustle6/sealeo/

cat hoothoot0/jolteon hoothoot0/totodile > quagsire9\_11

cat grotle5 > crustle6/blastoisegrotle

cp quagsire9 crustle6/sealeo

# Задание №4

**Задание** Используя команды cat, wc, ls, head, tail, echo, sort, grep выполнить в соответствии с вариантом задания поиск и фильтрацию файлов, каталогов и содержащихся в них данных.

**Выполнение**

1. Подсчитать количество строк содержимого файла grotle5, результат дописать в тот-же файл, подавить вывод ошибок доступа

**wc -l grotle5 >> grotle5 2> /dev/null**

1. Вывести рекурсивно список имен и атрибутов файлов в директории lab0, начинающихся на символ 's', список отсортировать по возрастанию даты изменения записи о файле, ошибки доступа перенаправить в файл в директории /tmp

**ls -Rl | grep '^[-].\* s[^ ]\*$' | sort -k7**

1. Вывести содержимое файлов: vanillish, gastrodon, slowking, stunky с номерами строк, строки отсортировать по имени z->a, ошибки доступа не подавлять и не перенаправлять

**cat vanillish gastrodon slowking | sort -r | cat -n**

1. Вывести содержимое файла ninetales2 с номерами строк, оставить только строки, заканчивающиеся на 't', регистр символов игнорировать, ошибки доступа не подавлять и не перенаправлять

**cat ninetales2 | grep -i 't$' | cat -n**

1. Вывести рекурсивно список имен и атрибутов файлов в директории lab0, заканчивающихся на символ 'n', список отсортировать по убыванию количества жестких ссылок, добавить вывод ошибок доступа в стандартный поток вывода

**ls -lR Lab0 | grep 'n$' | sort -k2nr 2> /dev/stdout**

1. Вывести рекурсивно список имен и атрибутов файлов в директории lab0, заканчивающихся на символ 'o', список отсортировать по возрастанию количества жестких ссылок, добавить вывод ошибок доступа в стандартный поток вывода

**ls -lR Lab0 | grep 'o$' | sort -k2n 2> /dev/stdout**

# Задание №5

**Задание**

Выполнить удаление файлов и каталогов при помощи команд rm и rmdir согласно варианту задания.

**Выполнение**

cd Lab0

rm -f quagsire9

cd hoothoot0

rm -f jolteon

rm -f totodileninetal\*

rm -f koffingquagsi\*

cd ..

rm -r boldore4

cd hoothoot0

rmdir abomasnow

cd ..

cd ..

# Ответы на вопросы

1) Переменное окружение

2) Что есть среда

3) Что такое окружение

4) Что такое оболочка

5) Чем отличается оболочка от терминала и консоли

6) Что такое терминал

7) Чё-то про переменные, инициализация

8) Чем идеологически отличается операционная система и прошивка

Прошивка не предоставляет права для выполнения и запуска кода.3

9) Прочитать про mmu, tlb, сегментно страничная виртуальная память

10) Жёсткая ссылка

1. **Переменное окружение** (или переменные окружения) — это набор значений, которые используются операционной системой и программами для настройки их работы. Эти переменные могут хранить информацию о конфигурации системы, пользователе, пути к исполняемым файлам и другим параметрам.

Определение: Переменные окружения представляют собой пары "имя-значение". Например, PATH — это переменная окружения, которая содержит список каталогов, в которых система ищет исполняемые файлы.

Использование: Программы могут использовать переменные окружения для получения информации о среде выполнения. Например, переменная HOME указывает на домашний каталог пользователя.

**Переменные окружения** — это пары "ключ-значение", которые хранят информацию о системе и пользователе. Например:

• PATH: Определяет, где система ищет исполняемые файлы.

• HOME: Указывает на домашнюю директорию пользователя.

• USER: Содержит имя текущего пользователя.

**Среда** **выполнения**

Это набор условий, в которых выполняется программа. Он может включать доступные библиотеки, настройки системы и другие параметры, которые влияют на поведение программы.

**Оболочка** (Shell) также считается частью среды, поскольку она предоставляет интерфейс для взаимодействия с операционной системой и управления переменными окружения.

Определение: Оболочка — это программа, которая предоставляет интерфейс для взаимодействия пользователя с операционной системой. Она интерпретирует команды, вводимые пользователем, и выполняет их.

• Примеры: Bash, Zsh, Fish, Tcsh.

• Функции: Оболочка позволяет запускать программы, управлять файлами и директориями, а также выполнять скрипты.

**Терминал (Terminal)**

• Определение: Терминал — это интерфейс (обычно графический), который позволяет пользователю взаимодействовать с оболочкой. Это может быть как физическое устройство (например, старый терминал), так и программное обеспечение (эмуляторы терминала).

• Примеры: GNOME Terminal, Konsole, xterm.

• Функции: Терминал отображает текстовый ввод и вывод, позволяя пользователю вводить команды и видеть результаты.

**Консоль (Console**)

• Определение: Консоль обычно относится к физическому устройству или виртуальному терминалу, который предоставляет прямой доступ к системе. В современных системах это может быть текстовый режи

м без графического интерфейса.

• Примеры: Консоль в режиме восстановления, виртуальные консоли (например, Ctrl + Alt + F1).

• Функции: Консоль позволяет пользователю взаимодействовать с системой напрямую, без необходимости в графическом интерфейсе.

▎**Взаимосвязь**

• Терминал служит интерфейсом для запуска оболочки.

• Оболочка обрабатывает команды, введенные пользователем через терминал.

• Консоль может быть местом, где работает оболочка, но не обязательно требует графического интерфейса.

В общем, терминал — это интерфейс для работы с оболочкой, а консоль — это более широкий термин, который может включать в себя как физические устройства, так и текстовые интерфейсы.

**Инициализация переменной**

В Bash переменные инициализируются без пробелов вокруг знака =:

my\_variable="Hello, World!

▎Использование переменной

Чтобы использовать значение переменной, нужно префиксировать её имя символом $:

echo $my\_variable

**MMU** (Memory Management Unit)

• MMU — это аппаратный компонент, который отвечает за управление доступом к памяти в компьютере. Основная задача MMU — преобразование виртуальных адресов, используемых программами, в физические адреса, которые используются для доступа к оперативной памяти.

• MMU также обеспечивает защиту памяти, предотвращая доступ одной программы к памяти другой программы и обеспечивая изоляцию процессов.

**TLB** (Translation Lookaside Buffer)

• TLB — это кэш, который используется MMU для ускорения преобразования виртуальных адресов в физические. Когда процесс обращается к памяти, MMU сначала проверяет TLB, чтобы увидеть, есть ли нужное соответствие виртуального и физического адреса

• Если соответствие найдено (TLB hit), преобразование происходит быстро. Если нет (TLB miss), MMU обращается к таблице страниц в основной памяти, что занимает больше времени.

**Сегментно-страничная виртуальная память**

• Сегментно-страничная виртуальная память — это метод управления памятью, который сочетает в себе сегментацию и страничную организацию памяти.

• Сегментация делит память на логические сегменты (например, код, данные, стек), что позволяет программам использовать более удобные структуры данных и облегчает управление памятью.

• Страничная организация разбивает память на фиксированные блоки (страницы), что упрощает управление памятью и уменьшает фрагментацию.

• В этом подходе каждый сегмент может быть разбит на страницы, и MMU использует таблицы сегментов и таблицы страниц для преобразования виртуальных адресов в физические.

Таким образом, MMU и TLB работают вместе для эффективного управления памятью, а сегментно-страничная система позволяет более гибко организовать использование памяти в зависимости от потребностей программ.

# Вывод по работе

В ходе работы я научился работать с командной строкой Linux, узнал базовые команды.