



МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт Информационных Технологий

Кафедра Вычислительной Техники

Практическая работа №3

по дисциплине

«Архитектура ВМиС»

Студент группы: **ИКБО-11-22**

Гришин А. В.
(Фамилия студента)

Преподаватель

Рыжова А.А.
(Фамилия преподавателя)

Москва 2023

Оглавление

Практическая работа номер №3.....	3
Вывод	6

Практическая работа номер №3.

Введение

AWK — это интерпретируемый скриптовый C-подобный язык построчного разбора и обработки входного потока (например, текстового файла) по заданным шаблонам (регулярным выражениям). Используется в bash (SH) скриптах. Благодаря AWK в нашем распоряжении оказывается язык программирования, а не довольно скромный набор команд, отдаваемых редактору. С помощью языка программирования AWK можно выполнять следующие действия:

- объявлять переменные для хранения данных;
- использовать арифметические и строковые операторы для работы с данными;
- использовать структурные элементы и управляющие конструкции языка, такие, как условные операторы и циклы;
- реализовать сложные алгоритмы обработки данных;
- создавать форматированные отчёты.

AWK может запоминать контекст, делать сравнения, создавать форматированные отчёты, которые удобно читать и анализировать. Это оказывается очень кстати при работе с лог-файлами, которые могут содержать миллионы записей. При надлежащей сноровке, она может объединять множество строк. Awk – это инструмент, предоставляющий несколько очень удобных способов обработки текстовых данных, которые могут пригодиться в повседневной жизни.

Выполнение работы

1. Вывод списка файлов, имеющих доступ для групп пользователей по чтению (Рисунок 1).

```
grishin@grishin-VirtualBox:~/grishin$ ls -RL | awk '{split($1, chars, ""); if (chars[5] == "r" && chars[1] != "d") print }'
```

Permissions	User	Group	Size	Month	Day	Time	File
-rw-rw-r--	1	grishin	209	дек	11	12:58	calendar.txt
-rw-rw-r--	1	grishin	316	дек	11	12:49	data.txt
-rw-rw-r--	1	grishin	102	дек	11	12:38	database1.txt
-rw-rw-r--	1	grishin	100	дек	11	12:42	database2.txt
-rw-rw-r--	1	grishin	59	дек	11	12:43	database3.txt
-rw-rw-r--	1	grishin	316	дек	11	12:57	filteredsorted.txt
-rw-rw-r--	1	grishin	79	дек	11	12:52	filtered.txt
-rw-rw-r--	1	grishin	24	дек	11	12:50	output.txt
-rw-rw-r--	1	grishin	316	дек	11	12:56	sorted.txt

```
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 1 – файлы, имеющие доступ по чтение

2. Вывод списка каталогов, имена которых состоят из английских букв. (См. Рис. 2)

```
grishin@grishin-VirtualBox:~/grishin$ ls -l | awk '$9~/[a-z]/ {print $9}'
```

Directory
database
temp

```
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 2 – Файлы с именами на английском

3. Определение количества байтов, занятых текстовыми файлами (txt) (Рисунок 3).

```
grishin@grishin-VirtualBox:~/grishin$ find . -type f -exec du -ch {} + -exec {} \;
```

File	Size
./database/data.txt	4,0K
./database/calendar.txt	4,0K
./temp/database3.txt	4,0K
./temp/report/filtered.txt	4,0K
./temp/report/sorted.txt	4,0K
./temp/report/filteredsorted.txt	4,0K
./temp/report/output.txt	4,0K
./temp/database1.txt	4,0K
./temp/database2.txt	4,0K
итого	36K

```
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 3 – Количество байтов

4. Определение количества блоков, содержащих текущий каталог (Рисунок 4).

```
grishin@grishin-VirtualBox:~/grishin$ ls -l | awk '{sum+=$2} END {print sum}'  
13  
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 4 – Количество блоков

5. Сортировка списка файлов текущего каталога по возможностям доступа (Рисунок 5).

```
grishin@grishin-VirtualBox:~/grishin$ ls -l | sort -k 1  
drwxrwxr-x 2 grishin grishin 4096 дек 11 12:58 database  
drwxrwxr-x 3 grishin grishin 4096 дек 11 12:45 temp  
итого 8  
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 5 – Сортировка по доступу

6. Вывод списка каталогов, в которых обнаружены файлы с определенным именем (Рисунок 6).

```
grishin@grishin-VirtualBox:~/grishin$ find . -name report  
./temp/report  
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 6 – Поиск файла

7. Подсчет количества вхождений пользователя в систему (Рисунок 7).

```
grishin@grishin-VirtualBox:~/grishin$ last | grep grishin | awk '{sum+=1} END {p  
rint sum}'  
1  
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 7 – Количества зафиксированных входов в систему

8. Вывод списка пользователей, отсортированного по времени входа в систему (Рисунок 8).

```
grishin@grishin-VirtualBox:~/grishin$ last | grep grishin | awk '{print $4, $5,  
$6, $7, $8, $9}' | sort -nk3  
Mon Dec 11 12:22 gone -  
grishin@grishin-VirtualBox:~/grishin$
```

Рисунок 8 – Сортировка по времени входа

Вывод

В данной практической работы мы познакомились с возможностями программируемого фильтра `awk`. Фильтр широко применяется для обработки данных и формирования различного вида отчетов. Для более глубокого изучения всех возможностей фильтра рекомендуется изучить справочные страницы по команде `awk`.