



ALTADM1.

**Администрирование ОС
Альт.**

Часть 1

Сборник практических заданий

Лабораторная работа 1

1. Результат выполнения лабораторных работ — это **три** сценария на языке командного интерпретатора.:
первый (основной) выполняет задание;
второй (тестовый) проверяет правильность работы основного; **третий** (обнуляющий) возвращает систему в исходное состояние так, чтобы можно было запустить процесс выполнения сценариев сначала.
2. Задача **основного** сценария первой лабораторной работы: создать файл **my_file**, записать в него с помощью соответствующих команд:
системную дату, текущий рабочий каталог, имя пользователя, название и версию ядра операционной системы.
3. **Тестовый** сценарий должен проверить наличие файла **my_file** и вывести его содержимое.
4. **Третий** сценарий удаляет файл **my_file** или сообщает об его отсутствии.
5. Для начала работы запустите текстовый редактор **nano**. Наберите шебанг и первые комментарии с описанием сценария:
#!/bin/sh
сценарий записи системной информации в файл
лабораторная работа № 1

Дальше следуют команды создания файла **my_file** и записи данных в него:

```
touch my_file

date > my_file

pwd >> my_file

who >> my_file

uname -sr >>
my_file
```

6. Сохраните файл, нажав **^+X** (Ctrl+X).
Имя файла первого скрипта первой работы должно быть **base1.sh**
7. Для того, чтобы можно было запустить скрипт, необходимо сделать файл исполняемым (наберите в командной строке, после выхода из редактора): **chmod +x ./base1.sh**
8. Попробуйте выполнить сценарий: **./base1.sh**
9. Отобразите на экране содержимое файла **my_file**, выполнив: **cat my_file**
10. Вернитесь к редактированию файла **base1.sh**, набрав: **nano base1.sh**

11. Обратите внимание на цветное оформление сценария.
Добавьте комментарии в строках создания и записи файла:

```
touch my_file      # создание файла

date > my_file     # запись даты и времени
pwd >> my_file     # запись текущего каталога
who >> my_file     # запись имени пользователя
uname >> my_file   # запись названия ОС
```

12. Сохраните изменения и закройте редактор.
13. Теперь создайте **тестовый** скрипт командой: `nano test1.sh`

Так же начните с шебанга и описания скрипта:

```
#!/bin/sh

# сценарий проверки правильности
выполнения # Лабораторной работы № 1

cat my_file # вывод на экран содержимого файла my_file
echo $?     # вывод кода завершения последней операции
```

14. Сохраните файл и закройте редактор.
15. Настройте права для выполнения и этого сценария: `chmod +x ./test1.sh`
16. Выполните сценарий:
 `./test1.sh`
17. Убедитесь, что код завершения (последнее число) равен **0**.
 Это говорит об **отсутствии ошибок** при выполнении команды.
18. Скопируйте файл `test1.sh` в новый файл `reset1.sh`. Затем вызовите его на редактирование:

```
cp test1.sh reset1.sh

nano reset1.sh
```

19. Измените комментарии и содержание редактируемого файла:

```
#!/bin/sh

# сценарий очистки результатов выполнения
# Лабораторной работы № 1

# удаление файла my_file
```

```
rm my_file && echo 'рабочий файл  
удалён'
```

```
echo $?          # вывод кода завершения последней операции
```

20. Сохраните файл и завершите редактирование. Как в предыдущих случаях сделайте

файл исполняемым: **chmod +x ./reset1.sh**

и запустите его

```
./reset1.sh
```

21. Можно проверить все три файла ещё одним последовательным запуском, а затем сверить с образцом содержание самих скриптов:

<i>base1.sh</i>	<pre>#!/bin/sh # сценарий записи системной информации в файл # Лабораторная работ № 1 touch my_file # создание файла date > my_file # запись даты и времени pwd >> my_file # запись текущего каталога в файл who >> my_file # запись имени в файл uname >> my_file # запись названия ОС в файл</pre>
<i>test1.sh</i>	<pre>#!/bin/sh # сценарий проверки правильности выполнения # Лабораторной работы № 1 cat my_file # вывод на экран содержимого файла my_file echo \$? # вывод кода завершения последней операции</pre>
<i>reset1.sh</i>	<pre>#!/bin/sh # сценарий очистки результатов выполнения # Лабораторной работы № 1 rm my_file && echo 'рабочий файл удалён' # удаление файла my_file echo \$? # вывод на экран кода завершения последней операции</pre>

Дополнительно :

В качестве самостоятельного задания разработайте три аналогичных сценария, при этом в файл **my_hist_file** должно быть записано 10 последних команд текущего сеанса работы.

Лабораторная работа 2

1. Результат выполнения лабораторных работ — это **три** сценария на языке командного интерпретатора.:
первый (основной) выполняет задание;
второй (тестовый) проверяет правильность работы основного; **третий** (обнуляющий) возвращает систему в исходное состояние так, чтобы можно было запустить процесс выполнения сценариев сначала.
2. Задача **основного** сценария **второй** лабораторной работы: создать в **/tmp** в новом файле **my_sys_2M.img** размером **2M** файловую систему **ext2**; провести монтирование новой ФС в **/mnt/new_disk**; исследовать возможность заполнения ФС в двух вариантах:
по количеству файлов;
по объёму ФС (дополнительное самостоятельное задание);
3. Тестовый сценарий должен проверить наличие ФС, показать её параметры, количество созданных файлов.
4. Для возврата к начальным условиям необходимо очистить и отмонтировать файловую систему, а затем удалить файл **my_sys_2M.img**.
5. Для начала работы запустите текстовый редактор **nano**: **nano base2.sh**
6. Наберите шебанг и первые комментарии с описанием сценария:

```
#!/bin/sh
```

```
# сценарий создания и монтирования ФС
```

```
# лабораторная работа № 2
```

Дальше следуют команды подготовки файла **my_sys_2M.img** и создания на нем

ФС:

```
truncate -s 2M /tmp/my_sys_2M.img      # файл заданного
```

размера

```
mkfs.ext2 /tmp/my_sys_2M.img           # созд. ФС ext2 mkdir
```

```
/mnt/new_disk                          # созд. точку монтирования mount
```

```
/tmp/my_sys_2M.img /mnt/new_disk       # монтируем ФС df -H
```

```
/mnt/new_disk                          # выводим сведения о ФС
```

Теперь приступаем к заполнению ФС:

```
mkdir /mnt/new_disk/new_dir           # создаем каталог в новой ФС
```

```
# в новом каталоге: 1000 файлов, причем сообщения об ошибках
cd /mnt/new_disk/new_dir
```

```
i=1; while touch file_$i; do echo file_$i; i=$((i+1)); done
```

(Предложите альтернативный вариант цикла, например с использованием **for**.)

7. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий с правами суперпользователя. После выполнения не забудьте завершить сеанс работы суперпользователя. **chmod +x ./base2.sh su ./base2.sh**
8. Начните работу с тестовым сценарием: **nano test2.sh**
9. Наберите шебанг и первые комментарии с описанием сценария:

```
#!/bin/sh
```

```
# сценарий исследования ФС
```

```
# лабораторная работа № 2
```

10. Тестовый сценарий должен вывести информацию о файловой системе и посчитать количество файлов в созданном каталоге:

```
df -i /mnt/new_disk # выводим сведения о количестве I-
нодов cd /mnt/new_disk/new_dir echo "количество созданных
файлов:" ls -l | wc -l # считаем количество файлов
```

11. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий.

```
chmod +x ./test2.sh
```

```
./test2.sh
```

12. Начните работу с обнуляющим сценарием:

```
nano reset2.sh
```

13. Наберите шебанг и первые комментарии с описанием сценария:

```
#!/bin/sh
```

```
# сценарий удаления ФС
```

```
# лабораторная работа № 2
```

14. Для возврата в исходное состояние нужно удалить созданные на новой ФС файлы и каталоги, размонтировать её, удалить файл самой системы и каталог в /mnt. Часть этих операций потребует прав суперпользователя:

```
cd /mnt/new_disk/ # переходим в ФС
```

```
rm -rf # удаляем файлы
```

```
umount /mnt/new_disk # размонтируем ФС
```

- ```
rm /tmp/my_sys_2M.img # удаляем файл my_sys_2M.img
```
15. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий и запустите сценарий с правами суперпользователя. После выполнения не забудьте завершить сеанс работы суперпользователя. **. chmod +x ./reset2.sh su ./reset2.sh**

---

**Дополнительно :**

В качестве самостоятельного задания разработайте три аналогичных сценария, но исследуйте переполнение файловой системы по объёму.

## Лабораторная работа 3

1. Как обычно, результат выполнения лабораторной работы — это **три** сценария на языке командного интерпретатора:  
**первый** (основной) выполняет задание; **второй** (тестовый) проверяет правильность работы основного; **третий** (обнуляющий) возвращает систему в исходное состояние так, чтобы можно было запустить процесс выполнения сценариев сначала.
2. Задача **основного** сценария третьей лабораторной работы:  
завести две локальные **групповые** учётные записи: **group1** и **group2**;  
создать три локальные **пользовательские** учётные записи: **user1**, **user2**, **user3**; включить пользователей в группы следующим образом:  
**первый** пользователь входит только в **первую** группу,  
**второй** пользователь — в **первую** и **вторую** группы,  
**третий** — только во **вторую**;  
создать в пределах ветки дерева каталогов **/srv** каталоги **dir1** и **dir2**;  
разграничить права следующим образом:  
подкаталог **dir1** доступен на запись только группе **group1**,  
**dir2** — только группе **group2**.
3. Тестовый сценарий должен проверить соответствие прав на файлы, создаваемые в каталогах **dir1** и **dir2**.
4. Для возврата к начальным условиям необходимо удалить пользователей и группы, а затем удалить созданные каталоги и файлы.
5. Для начала работы запустите текстовый редактор **nano**: **nano base3.sh**
6. Наберите шебанг и первые комментарии с описанием сценария:

```
#!/bin/sh
```

```
сценарий создания групп пользователей
лабораторная работа № 3
```

Дальше следуют команды создания групп и пользователей:

```
groupadd group1 # создаем группы
groupadd group2
```

```
adduser user1 # новые пользователи
adduser user2
adduser user3
gpasswd -a user1 group1 # добавляем пользователей в
группы
```



```
gpasswd -a user2 group1
```

```
gpasswd -a user2 group2
```

```
gpasswd -a user3 group2
```

```
for i in `seq 1 3` do # инфо о пользователях и группах
echo "пользователь user$i" id user$i done
```

```
mkdir /srv/dir1 /srv/dir2 # создаем каталоги
chgrp group1 dir1 # "отдаем" каталоги соотв. группам
chgrp group2 dir2
```

```
chmod 2775 /srv/dir1 /srv/dir2 # права на запись в
каталоги
```

```
ls -la /srv
```

7. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий и запустите сценарий с правами суперпользователя. После выполнения не забудьте завершить сеанс работы суперпользователя.
8. Разработайте тестовый сценарий (обратите внимание на различные варианты использования команды echo): nano test3.sh

```
#!/bin/sh
```

```
сценарий проверки прав пользователей
```

```
лабораторная работа № 3
```

```
su -l user2 -c ' cd /srv/dir1 touch file_u2d1
whoami && echo "создал(а) файл в каталоге" &&
pwd cd /srv/dir2 touch file_u2d2 echo "`whoami`
создал(а) файл в каталоге `pwd`" ' # user1 -
доступ к файлам в dir1 su -l user1 -c ' cd
/srv/dir1/ whoami && echo "доступ к файлу в
каталоге" && pwd mv /srv/dir1/file_u2d1
```

```
/srv/dir1/file_u1d1 echo $? # user1 - доступ к
файлам в dir2
```

```
cd /srv/dir2/ echo "`whoami` создал(а) файл в
каталоге `pwd`" mv /srv/dir1/file_u2d2
/srv/dir1/file_u1d2 echo $? '
```

9. Аналогично — для пользователя user3.

Затем сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий.

10. Разработайте обнуляющий сценарий:

```
nano reset3.sh
```

```
#!/bin/sh
```

```
сценарий удаления файлов, групп и пользователей
лабораторная работа № 3
```

```
удаление каталогов и файлов
```

```
rm -rf /srv/dir1 /srv/dir2
```

```
удаление групп и
пользователей
```

```
userdel -r user1
```

```
userdel -r user2
```

```
userdel -r user3
```

```
groupdel group1
```

```
groupdel group2
```

11. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий и запустите сценарий с правами суперпользователя. После выполнения не забудьте завершить сеанс работы суперпользователя.

---

**Дополнительно :**

В качестве самостоятельного задания разработайте три аналогичных сценария, при этом имена пользователей и названия групп должны задаваться как параметры сценария. Должна проводиться проверка количества передаваемых параметров.

## Лабораторная работа 4

1. В этой лабораторной работы нужно создать **четыре** сценария на языке командного интерпретатора.:  
**стартовый** (подготовительный) задает рабочую среду; **первый** (основной) выполняет задание; **второй** (тестовый) проверяет правильность работы основного; **третий** (обнуляющий) возвращает систему в исходное состояние так, чтобы можно было запустить процесс выполнения сценариев сначала.
2. Задача **стартового** сценария четвертой лабораторной работы: запустить в фоновом режиме 5 процессов;  
3 из которых являются **целевыми**, а 2 — **дополнительными**.  
Для примеров фоновых процессов можно выбрать `ping ya.ru` и `ping yandex.ru`.
3. **Основной** сценарий должен прекратить выполнение только **целевых** процессов.
4. **Тестовый** сценарий должен сообщать сколько целевых и дополнительных процессов выполняется в данный момент.
5. Для возврата к начальным условиям необходимо прекратить выполнение всех запущенных процессов.
6. В этой лабораторной работе первым разработайте **тестовый** сценарий. Запустите текстовый редактор: `nano test4.sh`
7. Наберите шебанг и первые комментарии с описанием сценария:

```
#!/bin/sh
```

```
сценарий мониторинга запущенных процессов
лабораторная работа № 4
```

```
echo "ping по адресу ya.ru" # количество основных процессов
pgrep -f ya.ru | wc -l
```

```
echo "ping по адресу yandex.ru" # количество доп. процессов
pgrep -f yandex.ru | wc -l
```

8. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий. При первом запуске в результате должны получиться нули.
9. Разработайте стартовый сценарий, : `nano start4.sh`

```
#!/bin/sh
```

```
сценарий запуска фоновых процессов
лабораторная работа № 4
```

```
запускаем 3 целевых процесса
```

```
for i in `seq 1 3`; do ping ya.ru >/dev/null &
done
```

```
запускаем 2 дополнительных процесса
```

```
for i in 1 2; do ping yandex.ru >/dev/null &
done
```

10. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий.
11. После этого снова запустите `test4.sh`, проверьте количество целевых и дополнительных процессов.
12. Основной сценарий должен «убить» процессы, связанные с `ya.ru`, и «пощадить» все остальные:  
`nano base4.sh`

```
#!/bin/sh
```

```
сценарий целевого прекращения процессов
```

```
лабораторная работа № 4
```

```
выбираем и прекращаем
процессы
```

```
for p in `pgrep -f ya.ru` do
kill $p done
```

13. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий. После чего запустите `test4.sh` и проверьте количество целевых и дополнительных процессов.
14. Очищающий сценарий прекращает все процессы, запущенные в ходе работы:  
`nano reset4.sh`

```
#!/bin/sh
```

```
сценарий очистки для
```

```
лабораторная работа № 4 # прекращаем
```

```
все процессы из Лаб.работы 4
```

```
for p in `pgrep -f
ya.ru` do kill $p
done
```

```
for p in `pgrep -f
yandex.ru` do kill $p
done
```

15. Сохраните файл и закройте редактор. Настройте права на исполнение и запустите сценарий. После чего запустите `test4.sh` и проверьте количество целевых и дополнительных процессов.

---

**Дополнительно :**

В качестве самостоятельного задания разработайте аналогичные сценарии, изменив их так, чтобы адреса сайтов (целевого и дополнительного) задавались как параметры при запуске сценария. Должна проводиться проверка количества передаваемых параметров.