

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 1. Файловые системы в Linux

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя root.

1.1. Работа с разделами дисков

Задание 1. Добавьте два новых диска в систему (размером 10 Гб каждый).

Выключите виртуальную машину. В VirtualBox: Файл → Настройки → Носители → контроллер SATA → «Добавить жёсткий диск» → «Создать новый диск» (тип VDI, динамический, размер 10 Гб). Аналогично добавьте второй диск. Запустите машину и проверьте, что диски появились:

```
lsblk
fdisk -l # должны появиться /dev/sdb и /dev/sdc
```

Задание 2. Создайте с помощью fdisk основной раздел размером 1 Гб на новом диске. Тип раздела 0x8e (LVM).

```
fdisk /dev/sdb
```

В интерактивном режиме выполните последовательность команд:

```
n → p (primary) → 1 → <Enter> (начало по умолчанию) → +1G
t → 8e # Linux LVM
w # записать таблицу разделов и выйти
```

Проверка:

```
fdisk -l /dev/sdb
lsblk -f
```

Задание 3. Создайте расширенный раздел на всё оставшееся место.

```
fdisk /dev/sdb
n → e (extended) → 2 → <Enter> (начало после основного) → <Enter> (весь объём) →
w
```

Расширенный раздел позволяет создавать внутри себя логические разделы (они получают номера 5, 6, ...).

Задание 4. Командой parted создайте логический раздел размером 1 Гб, установите тип раздела 0x83.

```
parted /dev/sdb
(parted) unit MiB
(parted) mkpart logical ext2 1025MiB 2049MiB
(parted) print
(parted) quit
```

В parted тип раздела определяется указываемой файловой системой (ext2/ext3/ext4 дают тип 0x83 Linux). Полученный логический раздел обычно имеет номер 5. Проверка: parted /dev/sdb print; fdisk -l /dev/sdb.

Задание 5. С помощью sfdisk скопируйте таблицу разделов с одного диска на другой.

```
sfdisk -d /dev/sdb | sfdisk /dev/sdc
```

Либо через промежуточный файл:

Глава 1. Файловые системы в Linux

```
sfdisk -d /dev/sdb > /tmp/tab.sdb
sfdisk /dev/sdc < /tmp/tab.sdb
sfdisk -d /dev/sdc # проверка копии
```

Диски должны быть одинакового размера. При возникновении ошибок используйте `sfdisk --force`.

1.2. Управление ФС ext*

Задание 1. Создайте файловую систему ext2 на первом логическом диске: 10 % для суперпользователя и метка тома logic1.

```
mkfs.ext2 -m 10 -L logic1 /dev/sdb5
```

Проверка:

```
dumpe2fs -h /dev/sdb5 | grep -E "Filesystem volume name|Reserved block count|Reserved blocks"
```

Номер логического раздела уточните по выводу `parted print`.

Задание 2. Проверьте файловую систему и поверхность логического диска.

```
e2fsck -f /dev/sdb5 # проверка структуры ФС
e2fsck -f -c /dev/sdb5 # -c — проверка поверхности (поиск плохих блоков)
```

Файловая система при проверке должна быть размонтирована.

Задание 3. Измените все файловые системы ext2 на ext3.

```
tune2fs -j /dev/sdb5 # добавляет журнал (признак has_journal)
```

Повторите для каждого ext2-раздела. Проверка:

```
tune2fs -l /dev/sdb5 | grep -i journal
blkid /dev/sdb5 # TYPE="ext3"
```

Задание 4. Установите на одном из логических дисков периодичность проверки, равную 1 году.

```
tune2fs -i 1y /dev/sdb5
tune2fs -l /dev/sdb5 | grep -iE "check interval|Time before next check"
```

Задание 5. Верните на одном из логических дисков файловую систему ext2 (уберите журнал).

```
umount /dev/sdb5 # если смонтирован
tune2fs -O ^has_journal /dev/sdb5
e2fsck -f /dev/sdb5
mount -t ext2 /dev/sdb5 /mnt
```

Проверка: `blkid /dev/sdb5 (TYPE="ext2")`. При сбое проще пересоздать ФС: `mkfs.ext2 /dev/sdb5`.

Задание 6. Создайте с помощью parted логический раздел 1 Гб и на нём файловую систему ext3.

```
parted /dev/sdb
(parted) mkpart logical ext3 3072MiB 4096MiB
(parted) quit
mkfs.ext3 /dev/sdb6
blkid /dev/sdb6
```

Задание 7. Увеличьте последний логический раздел на 1 Гб с помощью parted, затем увеличьте файловую систему.

```
parted /dev/sdb
(parted) resizepart 6 5120MiB # новое значение конца раздела
```

Глава 1. Файловые системы в Linux

```
(parted) quit
resize2fs /dev/sdb6
```

Если у вашей версии parted нет команды resizepart — удалите раздел и создайте его заново большего размера, затем выполните resize2fs. Проверка: parted /dev/sdb print; после монтирования — df -h.

Задание 8. Преобразуйте ext3 в ext4 и проверьте после этого файловую систему.

```
umount /dev/sdb6
tune2fs -O extent,uninit_bg,^huge_file /dev/sdb6
e2fsck -f /dev/sdb6 # обязательный запуск после смены признаков
```

Проверка: blkid /dev/sdb6 (TYPE="ext4"); fsck /dev/sdb6.

Задание 9. С помощью debugfs установите счётчик монтирований, равный 0.

```
debugfs -w -R "ssv mount_count 0" /dev/sdb5
dumpe2fs -h /dev/sdb5 | grep -i "mount count"
ssv — сокращение от set_super_value (можно писать полностью).
```

Задание 10. Измените метку тома на своё имя с помощью debugfs.

```
debugfs -w -R "ssv volume_name <ваше_имя>" /dev/sdb5
dumpe2fs -h /dev/sdb5 | grep "Filesystem volume name"
blkid /dev/sdb5
```

Задание 11. Создайте в домашнем каталоге подкаталог dir с файлами 1.txt, 2.txt, 3.txt.

```
mkdir ~/dir
touch ~/dir/1.txt ~/dir/2.txt ~/dir/3.txt
```

Задание 12. Установите атрибут a для 1.txt, i для 2.txt и s для 3.txt.

```
chattr +a ~/dir/1.txt
chattr +i ~/dir/2.txt
chattr +s ~/dir/3.txt
lsattr ~/dir
```

Задание 13. Попробуйте выполнить с файлами операции: добавить информацию, перезаписать содержимое, удалить файлы.

```
echo hi >> ~/dir/1.txt # можно — append-only разрешает добавление
echo hi > ~/dir/1.txt # нельзя: "Operation not permitted"
rm ~/dir/1.txt # нельзя
echo hi >> ~/dir/2.txt # нельзя (immutable — файл неизменяем)
cat > ~/dir/2.txt # нельзя
rm ~/dir/2.txt # нельзя
rm ~/dir/3.txt # разрешено; при удалении блоки затираются
```

Атрибут a — только добавление; i — файл нельзя изменять и удалять; s — безопасное удаление (блоки перезаписываются).

1.3. Другие ФС

Задание 1. Проверьте, включена ли поддержка XFS в ядре, и, если включена, создайте ФС XFS на одном из логических дисков.

```
grep xfs /proc/filesystems
ls /lib/modules/$(uname -r)/kernel/fs/xfs
dpkg -l xfsprogs || apt-get install -y xfsprogs
mkfs.xfs /dev/sdb6
```

Задание 2. Проверьте созданную ФС XFS.

```
mount /dev/sdb6 /mnt
```

Глава 1. Файловые системы в Linux

```
xfs_info /mnt # параметры смонтированной ФС
xfs_repair -n /dev/sdb6 # проверка без внесения изменений
xfs_db -r -c "sb -p" /dev/sdb6
```

Задание 3. Подготовьте iso-образ каталога /etc и проверьте его.

```
apt-get install -y genisoimage # при необходимости
genisoimage -o /tmp/etc.iso -R -J /etc
isoinfo -l -i /tmp/etc.iso # проверка содержимого образа
mkdir /mnt/iso
mount -o loop /tmp/etc.iso /mnt/iso
ls /mnt/iso && umount /mnt/iso
```

Задание 4. Проверьте наличие модуля FUSE.

```
grep fuse /proc/filesystems
lsmod | grep fuse
```

Если модуль отсутствует: `modprobe fuse`; при необходимости установить пакет fuse: `apt-get install -y fuse`.

Задание 5. Проверьте, установлен ли пакет NTFS-3G, если нет — установите.

```
dpkg -l ntfs-3g # проверить, установлен ли пакет
apt-get install -y ntfs-3g
ntfs-3g --version
```

Задание 6. Добавьте в виртуальную машину USB-диск.

В VirtualBox: Настройки → USB → включить контроллер USB 3.0 → добавить USB-устройство. Внутри машины:

```
lsusb # устройство видно
fdisk -l # диск появился как /dev/sdb
```

Задание 7. Создайте на этом диске один раздел на весь диск. Тип раздела 7.

```
fdisk /dev/sdb
n → p → 1 → <Enter> → <Enter> → t → 7 → w
Tun 7 — HPFS/NTFS/exFAT.
```

Задание 8. Создайте на разделе ФС NTFS, уменьшите её размер в половину и вновь увеличьте на весь раздел.

```
apt-get install -y ntfs-3g ntfsprogs # ntfsresize и mkfs.ntfs
mkfs.ntfs -Q /dev/sdb1 # быстрая разметка NTFS
ntfsresize --info /dev/sdb1 # узнать текущий размер
mount -t ntfs-3g /dev/sdb1 /mnt && df -h /mnt
umount /mnt
ntfsresize --size 5G /dev/sdb1 # уменьшение в половину (размер диска 10 Гб)
ntfsresize /dev/sdb1 # увеличение обратно до максимального размера
mount -t ntfs-3g /dev/sdb1 /mnt && df -h /mnt
```

1.4. Монтирование ФС

Задание 1. Создайте в каталоге /opt два подкаталога data1 и data2.

```
mkdir -p /opt/data1 /opt/data2
```

Задание 2. Смонтируйте один из разделов с опцией -n в каталог /opt/data1.

```
mount -n /dev/sdb1 /opt/data1
```

Опция -n запрещает запись в /etc/mtab. В современных системах /etc/mtab — ссылка на /proc/self/mounts, поэтому опция почти не влияет.

Задание 3. Командой mount проверьте список смонтированных устройств.

Глава 1. Файловые системы в Linux

mount

Задание 4. Смонтируйте этот же раздел в каталог /opt/data2.

```
mount /dev/sdb1 /opt/data2
```

Задание 5. Ещё раз проверьте список смонтированных устройств.

```
mount # один и тот же раздел смонтирован в две точки монтирования
```

Задание 6. Создайте в /etc/fstab запись для монтирования USB-диска по требованию пользователя в каталог /media/usb1. Используйте идентификатор /dev/disk/by-id/.

```
ls -l /dev/disk/by-id/ # найти идентификатор диска
```

Строка в /etc/fstab (пример):

```
/dev/disk/by-id/usb-VBOX_HARDDISK-0:0-part1 /media/usb1 ntfs-3g
noauto,user,nosuid,noexec,nodev,rw 0 0
mkdir -p /media/usb1
mount /media/usb1 # монтирование разрешено обычному пользователю
umount /media/usb1
```

Задание 7. Настройте автоматическое монтирование для CD-ROM.

Вариант через autofs (пакет autofs): создайте файл /etc/auto.master.d/cdrom.autofs с содержимым:

```
/cd /etc/auto.cdrom --timeout=10
```

и карту монтирования /etc/auto.cdrom:

```
cd -fstype=iso9660,ro,nosuid,nodev :/dev/sr0
systemctl restart autofs
```

Вставьте диск и обратитесь к каталогу: ls /cd/cd. В графическом окружении CD-ROM автоматически монтируется средствами gvfs/udisks2.

Задание 8. Настройте монтирование только одобренных USB-дисков посредством autofs.

1. Убедитесь, что в графической среде пользователи могут подключать USB-диски (пакет e2fsprogs, udisks2, environment gvfs).

2. Правило udev, скрывающее USB-диски от udisks, в файле /etc/udev/rules.d/80-udisk.rules:

```
cat > /etc/udev/rules.d/80-udisk.rules <<EOF
KERNEL=="sd*", ENV{ID_BUS}=="usb", ENV{UDISKS_IGNORE}="1"
EOF
udevadm control --reload
```

3. Закомментируйте в /etc/fstab строку монтирования USB-диска (добавьте # в начало строки).

4. Карта /etc/auto.usb (используйте ID своего диска):

```
cat > /etc/auto.usb <<EOF
flash1 -fstype=auto,nodev,nosuid,noexec :/dev/disk/by-id/usb-VBOX_HARDDISK-0:0-
part1
EOF
```

5. Файл /etc/auto.master.d/usb.autofs со следующим содержимым:

```
/usb /etc/auto.usb --timeout=60
```

6. Перезапустите autofs и проверьте доступ пользователей к каталогу:

```
systemctl restart autofs
ls /usb/flash1
```

Задание 9. Настройте автоматическое монтирование для SMB- и NFS-ресурсов (auto.smb, auto.net).

Прочитайте комментарии в начале файлов /etc/auto.smb и /etc/auto.net. Создайте файл

Глава 1. Файловые системы в Linux

/etc/auto.master.d/net.autofs:

```
/net /etc/auto.net --timeout=60
/smb /etc/auto.smb --timeout=60
systemctl restart autofs
```

Требуются пакеты: для NFS — `nfs-common` и `nfs-kernel-server` (сервер), для SMB — `samba-client` и `cifs-utils`. Доступ к ресурсам:

```
ls /net/<ip-адрес_сервера>
ls /smb/<имя_или_ip_сервера>
```

Задание 10. Подключитесь к указанным преподавателем сетевым папкам.

Уточните у преподавателя адреса сетевых папок (NFS или SMB). Вариант подключения NFS-ресурса:

```
apt-get install -y nfs-common
mkdir -p /mnt/teach
mount -t nfs <ip_преподавателя>:/path /mnt/teach
ls /mnt/teach
```

Вариант подключения SMB-ресурса:

```
apt-get install -y cifs-utils
mkdir -p /mnt/teach
mount -t cifs //<ip_преподавателя>/share /mnt/teach -o
username=<user>,uid=1000,gid=1000
ls /mnt/teach
```

1.5. ACL

Задание 1. Проверьте, что ядро поддерживает Linux ACL.

```
grep -i acl /boot/config-$(uname -r) # если файл конфигурации есть
zgrep CONFIG_FS_POSIX_ACL /proc/config.gz # вариант для ядер с /proc/config.gz
grep acl /proc/filesystems
apt-get install -y acl # утилиты setfacl/getfacl
getfacl --version
```

Задание 2. Измените `/etc/fstab` так, чтобы корневая файловая система монтировалась с ACL.

В `/etc/fstab` для корневого раздела добавьте опцию `acl`:

```
UUID=xxxx-xxxx-xxxx / ext4 defaults,acl 1 1
mount -o remount /
getfacl / # проверка
```

В современных ядрах *POSIX ACL* включены по умолчанию; опция *acl* оставлена для совместимости.

Задание 3. Создайте каталог `/data/project1`: владелец:группа `root:root`; группе `users` `rw`; пользователям `nobody` и `ftp` запретить доступ.

```
mkdir -p /data/project1
chown root:root /data/project1
setfacl -m g:users:rw /data/project1
setfacl -m u:nobody:--- /data/project1
setfacl -m u:ftp:--- /data/project1
getfacl /data/project1
```

Запрет для *nobody* и *ftp* реализуется установкой пустых прав (`---`). При необходимости установите маску: `setfacl -m m::rw /data/project1`.

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 2. Многодисковые накопители информации

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя root.

2.1. Система LVM

Задание 1. Для этой лабораторной работы лучше создать новую виртуальную машину. Добавьте 3 новых диска по 10 Гб.

В VirtualBox: Настройки → Носители → SATA → добавить три жёстких диска (VDI, динамический, 10 Гб). После загрузки диски видны как /dev/sdb, /dev/sdc, /dev/sdd. Проверка: lsblk.

Задание 2. Установите пакет lvm2, если он не установлен.

```
dpkg -l lvm2 || apt-get install -y lvm2
```

Задание 3. На первом диске создайте один раздел на весь диск. Тип 8e (MBR) или E6D6D379-F507-44C2-A23C-238F2A3DF928 (GPT).

```
fdisk /dev/sdb  
n → p → 1 → <Enter> → <Enter> → t → 8e → w
```

Вариант для GPT:

```
parted /dev/sdb mklabel gpt  
parted /dev/sdb mkpart primary 1MiB 100%  
parted /dev/sdb set 1 lvm on
```

Задание 4. Создайте физический том на созданном разделе и на втором диске.

```
pvccreate /dev/sdb1 /dev/sdc
```

Задание 5. Проверьте созданные LVM-диски командами pvscan, pvdisplay и pvs.

```
pvscan  
pvdisplay  
pvs
```

Задание 6. Создайте группу томов vg1 из одного раздела.

```
vgcreate vg1 /dev/sdb1
```

Задание 7. Получите информацию о группах командами vgdisplay и vgs.

```
vgdisplay  
vgs
```

Задание 8. Расширьте группу, добавив в неё второй диск.

```
vgextend vg1 /dev/sdc
```

Задание 9. С помощью vgs -o help получите данные о группе: имя, размер, свободное место и количество логических томов.

```
vgs -o help # список доступных полей  
vgs -o vg_name,vg_size,vg_free,lv_count
```

Задание 10. Создайте линейный логический том размером 1 Гб с ФС ext4. Затем увеличьте размер тома и ФС на 1 Гб.

```
lvcreate -L 1G -n vol1 vg1  
mkfs.ext4 /dev/vg1/vol1  
lvdisplay /dev/vg1/vol1
```

Глава 2. Многодисковые накопители информации

```
lvextend -L +1G /dev/vg1/vol1
resize2fs /dev/vg1/vol1 # увеличение ФС ext4
ФС ext4 увеличивается на смонтированном томе.
```

Задание 11. Определите командой `lvdisplay -m`, какой диск используется для этого тома.

```
lvdisplay -m /dev/vg1/vol1
```

Задание 12. Создайте ещё один логический том размером 1 Гб, но так, чтобы использовался второй диск.

```
lvcreate -L 1G -n vol2 vg1 /dev/sdc
lvdisplay -m /dev/vg1/vol2
```

Задание 13. Создайте зеркальный логический том размером 2 Гб с ФС XFS.

```
lvcreate -L 2G -n vol3 -m1 vg1 # зеркало (2 копии данных)
mkfs.xfs /dev/vg1/vol3
```

Задание 14. Смонтируйте тома в каталоги `/vol1`, `/vol2` и `/vol3` и сделайте монтирование постоянным.

```
mkdir -p /vol1 /vol2 /vol3
mount /dev/vg1/vol1 /vol1
mount /dev/vg1/vol2 /vol2
mount /dev/vg1/vol3 /vol3
blkid /dev/vg1/vol1 /dev/vg1/vol2 /dev/vg1/vol3
```

Запишите в `/etc/fstab` три строки вида:

```
UUID=<...vol1...> /vol1 ext4 defaults 0 0
UUID=<...vol2...> /vol2 ext4 defaults 0 0
UUID=<...vol3...> /vol3 xfs defaults 0 0
mount -a
```

Задание 15. Попробуйте удалить раздел из группы — вы получите ошибку, что диск занят.

```
vgreduce vg1 /dev/sdc
# Ошибка: Physical volume "/dev/sdc" still in use
```

Задание 16. Расширьте группу ещё на один диск и перенесите на него данные с раздела.

```
pvcreeate /dev/sdd
vgextend vg1 /dev/sdd
pvmove /dev/sdc # перенос всех экстендов с /dev/sdc на свободные физические тома
```

Задание 17. Теперь удалите раздел из группы — удаление должно пройти успешно.

```
vgreduce vg1 /dev/sdc
```

Задание 18. Размонтируйте тома и удалите их, удалите группу и физические тома.

```
umount /vol1 /vol2 /vol3
lvremove /dev/vg1/vol1 /dev/vg1/vol2 /dev/vg1/vol3
vgremove vg1
pvremove /dev/sdb1 /dev/sdc /dev/sdd
```

2.2. RAID

Задание 1. Установите пакет `mdadm`, если он не установлен.

```
dpkg -l mdadm || apt-get install -y mdadm
```

Задание 2. Удалите LVM-раздел и создайте три раздела: 2 Гб, 3 Гб и 5 Гб. Установите тип разделов Linux RAID.

Сначала освободите диск от LVM (если использовался): `pvremove /dev/sdb1`. Затем:

```
fdisk /dev/sdb
```

Глава 2. Многодисковые накопители информации

```
d # удалить существующие разделы
n → p → 1 → <Enter> → +2G ; t → 1 → fd
n → p → 2 → <Enter> → +3G ; t → 2 → fd
n → p → 3 → <Enter> → <Enter> ; t → 3 → fd
w
```

```
fdisk -l /dev/sdb # проверка
```

Задание 3. Скопируйте таблицу разделов с первого диска на второй и третий.

```
sfdisk -d /dev/sdb | sfdisk /dev/sdc
sfdisk -d /dev/sdb | sfdisk /dev/sdd
```

Задание 4. Создайте RAID 0 из трёх разделов по 2 Гб (итого 6 Гб) под swap. Подключите в /etc/fstab по UUID.

```
mdadm --create /dev/md0 --level=0 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1
mkswap /dev/md0
swapon /dev/md0
blkid /dev/md0
```

Строка в /etc/fstab:

```
UUID=<...md0...> swap swap sw 0 0
```

Зафиксируйте массив в конфигурации Debian (/etc/mdadm/mdadm.conf) и обновите initramfs:

```
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
update-initramfs -u
```

Задание 5. Создайте RAID 1 из разделов по 3 Гб, один диск запасной. ФС ext4, точка монтирования /opt/app1/data.

```
mdadm --create /dev/md1 --level=1 --raid-devices=2 --spare-devices=1
/dev/sdb2 /dev/sdc2 /dev/sdd2
mkfs.ext4 /dev/md1
mkdir -p /opt/app1/data
mount /dev/md1 /opt/app1/data
```

Задание 6. Создайте RAID 5 из разделов по 5 Гб с ФС XFS. Точка монтирования /opt/app1/log.

```
mdadm --create /dev/md2 --level=5 --raid-devices=3 /dev/sdb3 /dev/sdc3 /dev/sdd3
mkfs.xfs /dev/md2
mkdir -p /opt/app1/log
mount /dev/md2 /opt/app1/log
```

Задание 7. Опишите монтирование /opt/app1/data и /opt/app1/log в /etc/fstab по UUID. Перезагрузите систему и убедитесь, что массивы подключились.

```
blkid /dev/md1 /dev/md2
```

В /etc/fstab:

```
UUID=<...md1...> /opt/app1/data ext4 defaults 0 0
UUID=<...md2...> /opt/app1/log xfs defaults 0 0
```

Зафиксируйте все массивы:

```
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
update-initramfs -u
reboot
```

После перезагрузки проверьте: `df -h` и `cat /proc/mdstat` — все массивы (md0, md1, md2) должны быть активны.

Задание 8. Проверьте состояние созданных массивов.

```
cat /proc/mdstat
mdadm --detail /dev/md1
mdadm --detail /dev/md2
```

Задание 9. Пометьте один из активных разделов в зеркале как сбойный и проверьте

Глава 2. Многодисковые накопители информации

синхронизацию.

```
mdadm /dev/md1 --fail /dev/sdc2
cat /proc/mdstat # на запасном разделе /dev/sdd2 начался rebuild
mdadm --detail /dev/md1 | tail -20
```

Убедитесь, что в поле Rebuild Status идёт синхронизация данных на запасной диск.

Задание 10. Удалите помеченный как сбойный раздел из массива и вновь добавьте его как запасной.

```
mdadm /dev/md1 --remove /dev/sdc2
mdadm /dev/md1 --add /dev/sdc2
cat /proc/mdstat
mdadm --detail /dev/md1
```

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 3. BTRFS

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя root.

3.1. BTRFS

Задание 1. Создайте том BTRFS из двух дисков, в котором метаданные дублируются, а блоки данных чередуются.

Установите BTRFS-инструменты и пересоздайте разделы на двух дисках (например по 2 Гб) или используйте свободные:

```
apt-get install -y btrfs-progs
mkfs.btrfs -m raid1 -d raid0 /dev/sdb1 /dev/sdc1
mkdir /btrfs
mount /dev/sdb1 /btrfs
btrfs filesystem show /btrfs
```

Задание 2. Создайте два subvolume с именами sub1 и sub2.

```
btrfs subvolume create /btrfs/sub1
btrfs subvolume create /btrfs/sub2
btrfs subvolume list /btrfs
```

Задание 3. Установите общее ограничение 100 МБ на sub1 и sub2.

Используем группы квот (qgroup) — включаем оба subvolume в общую родительскую группу и задаём ей общий лимит:

```
btrfs quota enable /btrfs
btrfs qgroup create 1/0 /btrfs
btrfs subvolume list /btrfs # узнать ID: sub1=257, sub2=258
btrfs qgroup assign 0/257 1/0 /btrfs
btrfs qgroup assign 0/258 1/0 /btrfs
btrfs qgroup limit 100M 1/0 /btrfs
btrfs qgroup show -pcre /btrfs
```

Задание 4. Для sub1 установите общее ограничение 60 МБ.

```
btrfs qgroup limit 60M 0/257 /btrfs
btrfs qgroup show -pcre /btrfs
```

Задание 5. Для sub2 установите общее ограничение 80 МБ.

Глава 3. BTRFS

```
btrfs qgroup limit 80M 0/258 /btrfs
btrfs qgroup show -re /btrfs
```

Задание 6. Проверьте работу квот.

```
dd if=/dev/zero of=/btrfs/sub1/big.bin bs=1M count=100
```

При превышении лимита 60 МБ команда завершится с ошибкой «Disk quota exceeded» / «No space left on device». Текущее использование проверяется командами `btrfs qgroup show -re` и `btrfs qgroup show -pcre`.

Задание 7. Создайте резервную копию sub1. Удалите и восстановите sub1.

Резервной копией subvolume служит снимок (snapshot):

```
btrfs subvolume snapshot /btrfs/sub1 /btrfs/sub1.snap
```

Удаление:

```
btrfs subvolume delete /btrfs/sub1
```

Восстановление из снимка:

```
btrfs subvolume snapshot /btrfs/sub1.snap /btrfs/sub1
btrfs subvolume list /btrfs
```

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 4. Разделение ресурсов с помощью SAMBA

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя root.

4.1. SAMBA

Задание 1. Определите версию SAMBA, установленную в системе.

```
smbd --version
dpkg -l samba
```

Задание 2. Настройте SAMBA в качестве члена рабочей группы WORKGROUP.

Установите пакеты и раздел [global] в /etc/samba/smb.conf:

```
apt-get install -y samba samba-client
[global]
workgroup = WORKGROUP
security = user
server role = standalone server
passdb backend = tdbsam
testparm
systemctl enable --now smbd nmbd
```

Задание 3. Создайте домашние папки для пользователей на сервере.

Домашние каталоги создаются командой `useradd -m`. В `smb.conf` добавьте секцию:

```
[homes]
comment = Home Directories
browseable = no
read only = no
valid users = %S
```

Задание 4. Создайте пользователей user1, user2 и user3 в Linux и в SAMBA, проверьте доступ к домашним папкам.

Глава 4. Разделение ресурсов с помощью SAMBA

```
useradd -m user1; useradd -m user2; useradd -m user3
smbpasswd -a user1
smbpasswd -a user2
smbpasswd -a user3
```

Проверка доступа к собственным домашним папкам с клиентской машины:

```
smbclient -L //server -U user1
smbclient //server/user1 -U user1
```

Задание 5. Создайте общую папку SHARE с физическим расположением /srv/share.

```
mkdir -p /srv/share
```

Задание 6. Создайте группы ro_share и rw_share и с помощью ACL предоставьте доступ к /srv/share: чтение для ro_share, чтение и запись для rw_share. Убедитесь, что ACL наследуются новыми объектами.

```
groupadd ro_share; groupadd rw_share
setfacl -m g:rw_share:rwx /srv/share
setfacl -m g:ro_share:r-x /srv/share
```

Добавьте default-записи для наследования ACL новыми объектами:

```
setfacl -m d:g:rw_share:rwx /srv/share
setfacl -m d:g:ro_share:r-x /srv/share
```

Проверка:

```
getfacl /srv/share
touch /srv/share/test && getfacl /srv/share/test
```

Задание 7. Добавьте пользователей в группы: user1 — rw_share; user2 — ro_share; user3 никуда.

```
usermod -aG rw_share user1
usermod -aG ro_share user2
```

Задание 8. Настройте доступ к SHARE для групп: rw — для rw_share, только чтение — для ro_share.

Секция в /etc/samba/smb.conf:

```
[SHARE]
path = /srv/share
browseable = yes
read only = yes
read list = @ro_share
write list = @rw_share
valid users = @ro_share,@rw_share
testparm && systemctl restart smbd nmbd
```

Задание 9. Проверьте с другой машины доступ пользователей к общей папке.

```
smbclient //server/SHARE -U user2 # только чтение
smbclient //server/SHARE -U user1 # можно создавать файлы (put)
```

Задание 10 (*). Настройте один из компьютеров в качестве PDC для домена domn.

В /etc/samba/smb.conf на сервере:

```
[global]
workgroup = domn
server role = classic primary domain controller
security = user
domain logons = yes
logon path = \\%L\profiles\%U
[netlogon]
path = /var/lib/samba/netlogon
read only = yes
```

Глава 4. Разделение ресурсов с помощью SAMBA

Затем: `mkdir /var/lib/samba/netlogon; testparm; systemctl restart smbd nmbd`.

Задание 11 (*). Настройте вторую машину как член домена SAMBA.

На клиенте в `/etc/samba/smb.conf`:

```
[global]
workgroup = domn
security = user
server role = member server
```

Присоединение к домену:

```
net join member -U Administrator%пароль # или: net ads join -U Administrator
```

Проверка: `net ads testjoin; wbinfo -u`.

Задание 12 (*). Настройте один из компьютеров в качестве WINS-сервера, а второй — его клиентом.

На сервере в `[global]`:

```
wins support = yes
```

На клиенте в `[global]`:

```
wins server = <IP_сервера>
```

Затем: `testparm; systemctl restart nmbd`; проверка: `nmblookup -U <IP_сервера> <имя_компьютера>`.

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 5. Поддержка системных журналов

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя `root`.

5.1. Rsyslog

Задание 1. Настройте на первой машине сервер `rsyslog` на приём сообщений из сети. Журналы — в подкаталогах для каждого клиента в `/var/log/netclients/`, имя журнала соответствует `facility`.

В Debian `rsyslog` установлен по умолчанию (пакет `rsyslog`). Создайте каталог и файл `/etc/rsyslog.d/netlog.conf`:

```
mkdir -p /var/log/netclients
cat > /etc/rsyslog.d/netlog.conf <<'EOF'
module(load="imudp")
input(type="imudp" port="514")
$template ClientLog, "/var/log/netclients/%HOSTNAME%/%programname%.log"
*. * ?ClientLog
& stop
EOF
systemctl restart rsyslog
```

При необходимости откройте порт в межсетевом экране: `ufw allow 514/udp` или `iptables -A INPUT -p udp --dport 514 -j ACCEPT`. Для приёма по TCP добавьте `module(load="imtcp")` и `input(type="imtcp" port="514")`.

Задание 2. Настройте на второй машине передачу всех сообщений от пользователя

(facility user) на сервер И запись в файл /var/log/usermes.

В /etc/rsyslog.conf (или /etc/rsyslog.d/):

```
user.* /var/log/usermes
user.* @<IP_сервера> # @ — по UDP; @@ — по TCP
systemctl restart rsyslog
```

Задание 3. Проверьте работу rsyslog с помощью logger.

На клиенте отправьте сообщение:

```
logger -p user.notice "Тестовое сообщение"
```

На сервере проверьте, что сообщение появилось в файле соответствующего клиента:

```
ls -R /var/log/netclients
cat /var/log/netclients/<имя_клиента>/*.log
```

Задание 4. Настройте ротацию журналов, поступающих из сети.

Файл /etc/logrotate.d/netclients (пакет logrotate):

```
/var/log/netclients/*.log {
daily
rotate 7
compress
missingok
notifempty
copytruncate
}
```

Проверка: logrotate -d /etc/logrotate.d/netclients (режим отладки).

5.2. syslog-ng

Задание 1. Установите пакет syslog-ng на первую машину.

```
apt-get install -y syslog-ng
systemctl stop rsyslog && systemctl disable rsyslog
systemctl enable --now syslog-ng
```

Задание 2. Разрешите в опциях автоматическое создание каталогов.

В /etc/syslog-ng/syslog-ng.conf в блоке options:

```
options {
create_dirs(yes);
};
systemctl restart syslog-ng
```

Задание 3. Настройте в отдельном файле конфигурации приём сообщений от сетевых клиентов и запись в файлы, аналогично заданию по rsyslog. События записывать в каталог /var/log/syslog-ng/, порт UDP — 10514.

Файл /etc/syslog-ng/conf.d/net.conf:

```
source s_net {
udp(ip(0.0.0.0) port(10514));
};

destination d_net {
file("/var/log/syslog-ng/${HOST}/${PROGRAM}.log" create_dirs(yes));
};

log {
```

Глава 5. Поддержка системных журналов

```
source(s_net);
destination(d_net);
};
systemctl restart syslog-ng
```

Вместо `${PROGRAM}` для записи по facility можно использовать `${FACILITY}` и шаблон `file("/var/log/syslog-ng/${HOST}/${FACILITY}.log")`.

Задание 4. Настройте передачу всех сообщений на второй машине серверу syslog-ng.

В `/etc/syslog-ng/syslog-ng.conf` на клиенте:

```
destination d_sng {
    udp("<IP_сервера>" port(10514));
};
```

```
log {
    source(s_sys);
    destination(d_sng);
};
systemctl restart syslog-ng
```

Проверка: `logger -p user.notice "test"`; на сервере посмотреть файл `/var/log/syslog-ng/<имя_клиента>/user.log`.

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 6. Резервное копирование

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя root.

6.1. Инкрементальное копирование

Задание 1. Выполните полное и инкрементальное архивирование tar -g, восстановление, ответьте на вопрос о восстановлении.

Подготовка каталога:

```
mkdir dir1
for f in 1 2 3; do echo 1 > dir1/$f; done
```

Полное резервное копирование (создание архива `dir1.1.tar`):

```
tar -g snapshot.txt -cvf dir1.1.tar dir1
```

Замените в файлах 2 и 3 цифру 1 на 2:

```
echo 2 > dir1/2; echo 2 > dir1/3
```

Инкрементальное архивирование (`dir1.2.tar`):

```
tar -g snapshot.txt -cvf dir1.2.tar dir1
```

Замените в файле 3 цифру 2 на 3:

```
echo 3 > dir1/3
```

Инкрементальное архивирование (`dir1.3.tar`):

```
tar -g snapshot.txt -cvf dir1.3.tar dir1
```

Проверьте содержимое архивов:

```
tar -tvf dir1.1.tar
tar -tvf dir1.2.tar
```

Глава 6. Резервное копирование

```
tar -tvf dir1.3.tar
```

Удалите каталог `dir1` и восстановите данные из `dir1.1.tar` и `dir1.3.tar`:

```
rm -rf dir1
tar -g snapshot.txt -xvf dir1.1.tar
tar -g snapshot.txt -xvf dir1.3.tar
cat dir1/1 dir1/2 dir1/3
```

Результат: файл 1 = «1», файл 2 = «1» (архив `dir1.2.tar` с его изменением пропущен), файл 3 = «3».

Чтобы получить данные в нужном виде (1 = «1», 2 = «2», 3 = «3»), необходимо восстановить ВСЮ цепочку архивов в порядке возрастания:

```
rm -rf dir1
tar -g snapshot.txt -xvf dir1.1.tar
tar -g snapshot.txt -xvf dir1.2.tar
tar -g snapshot.txt -xvf dir1.3.tar
```

Задание 2. Сделайте то же самое, но с применением разностного архивирования.

Разностное архивирование: каждый следующий архив содержит все изменения относительно полного. Полный бэкап:

```
tar -g snap_full.txt -cvf dir1.1.tar dir1
```

Изменения (файлы 2 и 3 → 2):

```
echo 2 > dir1/2; echo 2 > dir1/3
```

Разностный архив — используем копию состояния полного бэкапа (чтобы изменения считались от полного):

```
cp snap_full.txt snap.d1.txt
tar -g snap.d1.txt -cvf dir1.d1.tar dir1
```

Изменения (файл 3 → 3):

```
echo 3 > dir1/3
```

Второй разностный архив:

```
cp snap_full.txt snap.d2.txt
tar -g snap.d2.txt -cvf dir1.d2.tar dir1
```

Восстановление: полный бэкап + последний разностный:

```
rm -rf dir1
tar -xvf dir1.1.tar
tar -g snap_full.txt -xvf dir1.d2.tar
cat dir1/1 dir1/2 dir1/3 # 1 = «1», 2 = «2», 3 = «3»
```

Преимущество разностного способа: для восстановления нужны всего два архива (полный и последний разностный).

Задание 3 (*). Используя `find` и `cpio`, сконструируйте команду для инкрементального копирования `cpio`.

Подсказка: `find` умеет сравнивать время с помощью `-mtime` (время, прошедшее после модификации) или `-newer <файл>` (сравнение с файлом-маркером).

Вариант командой:

```
find dir1 -mtime -1 -print | cpio -o -v > dir1.inc.cpio
```

либо с маркером:

```
find dir1 -newer /tmp/marker -print | cpio -o -v > dir1.inc.cpio
touch /tmp/marker
```

Задание 4 (*). Проверьте эту команду для сценария из задания по `tar`.

Глава 6. Резервное копирование

```
mkdir dir1; for f in 1 2 3; do echo 1 > dir1/$f; done
```

Полная копия:

```
find dir1 -print | cpio -o > dir1.full.cpio
```

изменения:

```
echo 2 > dir1/2; echo 2 > dir1/3  
find dir1 -mtime -1 -print | cpio -o > dir1.inc.cpio
```

восстановление полной копии и поверх инкрементальной:

```
rm -rf dir1  
cpio -idm < dir1.full.cpio  
cpio -idm < dir1.inc.cpio
```

6.2. dump/restore

Задание 1. Создайте архивную копию домашних каталогов пользователей. Удалите домашний каталог user2 и восстановите его.

```
tar -czvf /root/home.tar.gz /home  
rm -rf /home/user2
```

Восстановление:

```
cd / && tar -xzipvf /root/home.tar.gz  
ls -la /home/user2
```

Задание 2. Проверьте возможность работы dump на файловой системе XFS.

Утилита dump не поддерживает XFS — для XFS применяется механизм xfsdump/xfsrestore. Попытка запустить классический dump завершится ошибкой:

```
dump -0uf - /dev/md2 # ошибка: ФС XFS не поддерживается
```

Для XFS используйте пакет xfsdump:

```
apt-get install -y xfsdump  
xfs_freeze -f /opt/app1/log  
xfsdump -0 -f /tmp/xfs.bak /opt/app1/log  
xfs_freeze -u /opt/app1/log
```

Задание 3 (*). С помощью dump и restore перенесите данные из каталога /var в чередующийся том (RAID0).

```
apt-get install -y dump # пакет dump для ext-разделов
```

Сохраните содержимое /var в RAID-массив /dev/md0 (создан в главе 2). Вариант с dump, если под /var выделен отдельный раздел:

```
mkdir -p /mnt/target  
mkfs.ext4 /dev/md0 && mount /dev/md0 /mnt/target  
dump -0uf - /dev/<раздел_с_/var> | (cd /mnt/target && restore -rf -)
```

Если /var расположен в корневой ФС, используйте tar-аналог:

```
tar -C /var -cf - . | (cd /mnt/target && tar -xpf -)
```

Восстановление по требованию: umount /dev/md0; mount /dev/md0 /mnt/target; cd /var && restore -rf /mnt/target (либо скопировать данные tar'ом).

6.3. AMANDA

Задание 1. Настройте одну машину сервером AMANDA (архивация на диск), другую — клиентом. Установите amanda-server и amanda-client на сервер.

```
apt-get install -y amanda-server amanda-client
```

Глава 6. Резервное копирование

В Debian пакет устанавливает общего пользователя backup и каталог конфигурации /etc/amanda/.

Задание 2. На клиентах установите и настройте запуск клиента (amanda-client).

```
apt-get install -y amanda-client amanda-common
```

Разрешите серверу доступ к клиенту (файл /etc/amandahosts), на каждой клиентской машине:

```
echo "<IP_сервера_amanda> backup amdump" >> /etc/amandahosts
```

В Debian клиент запускается через xinetd/inetd (или службу amanda-client), например:

```
apt-get install -y xinetd
systemctl restart xinetd
```

Задание 3. Создайте новый набор. Имя набора определите самостоятельно (например mybackup).

```
mkdir -p /etc/amanda/mybackup
```

Задание 4. Добавьте на сервер новый диск 20 Гб, 1 раздел на весь диск, ФС ext4. Настройте /etc/fstab на монтирование в /amandastore.

```
fdisk /dev/sde # создать 1 раздел на весь диск
mkfs.ext4 /dev/sde1
mkdir -p /amandastore
blkid /dev/sde1
echo "UUID=<...> /amandastore ext4 defaults 0 0" >> /etc/fstab
mount -a && df -h /amandastore
```

Задание 5. В наборе создайте основной файл конфигурации amanda.conf. Количество циклов 5, период 1 неделя. Рассчитайте объем одного слота: $V_{\text{диска}} \cdot 0.9 / N_{\text{циклов}}$.

Объем слота: $20 \text{ Гб} \times 0.9 / 5 = 3.6 \text{ Гб}$ (≈ 3686 Мбайт). Пример /etc/amanda/mybackup/amanda.conf:

```
org "IT Cloud"
mailto "root@localhost"
dumpuser "backup"
tapecycle 6
runtapes 1
runspercycle 5
dumpcycle 1 week
tpchanger "amandastore"
tapedev "file:/amandastore/mybackup/vtapes"
tapetype DISK
labelstr "^MYBACKUP-[0-9]{4}$"
autolabel "MYBACKUP-%%%"
inparallel 1

define tapetype DISK {
    length 3686 mbytes
}

define changer amandastore {
    tpchanger "chg-disk:/amandastore/mybackup/vtapes"
    property "num-slot" "6"
    property "auto-create-slot" "yes"
}

define dumptype global {
    comment "Global definitions"
    # holdingdisk never
}
```

Глава 6. Резервное копирование

```
define dumptype gen {
    global
    program "GNUTAR"
    comment "generic"
    index yes
    compress fast
}

holdingdisk hd1 {
    directory "/amandastore/mybackup/hd1"
    use 5000 Mb
}

infofile "/amandastore/mybackup/curinfo"
logdir "/amandastore/mybackup/logs"
indexdir "/amandastore/mybackup/index"
```

Создайте каталоги, которые присутствуют в конфигурации!

```
sudo mkdir -p /amandastore/mybackup
sudo chown backup:backup /amandastore/mybackup
sudo -u backup mkdir -p /amandastore/mybackup/hd1
sudo -u backup mkdir -p /amandastore/mybackup/vtapes
sudo -u backup mkdir -p /amandastore/mybackup/logs
sudo -u backup mkdir -p /amandastore/mybackup/index
sudo -u backup mkdir -p /amandastore/mybackup/curinfo
```

Задание 6. Настройте архивацию каталогов /etc на сервере и клиентах в файле disklist.

Файл /etc/amanda/mybackup/disklist:

```
server-hostname /etc gen
client-hostname /etc gen
```

Замените имена на адреса/имена ваших машин.

Задание 7. Инициализируйте хранилище.

```
sudo -u backup amtape mybackup reset
sudo -u backup amtape mybackup inventory
```

Задание 8. Пометьте слоты.

```
sudo -u backup amtape mybackup slot 1 label MYBACKUP-0001
sudo -u backup amtape mybackup slot 2 label MYBACKUP-0002
sudo -u backup amtape mybackup slot 3 label MYBACKUP-0003
sudo -u backup amtape mybackup slot 4 label MYBACKUP-0004
sudo -u backup amtape mybackup slot 5 label MYBACKUP-0005
sudo -u backup amtape mybackup slot 6 label MYBACKUP-0006
sudo -u backup amtape mybackup inventory
```

Задание 9. Проверьте правильность конфигурации.

```
sudo -u backup amcheck mybackup
```

Задание 10. Настройте в планировщике запуск заданий раз в сутки. Время запуска 03:15.

```
echo "15 3 * * * root /usr/sbin/amdump mybackup >>
/var/log/amanda/mybackup/amdump.log 2>&1" >> /etc/crontab
```

Задание 11. Запустите архивацию вручную и проверьте результат.

```
sudo -u backup amdump mybackup
sudo -u backup amreport mybackup
sudo -u backup amadmin mybackup tape
```

Глава 6. Резервное копирование

```
sudo -u backup amcheckdump mybackup
```

Проверьте каталог /amandastore, отчёт и журнал /amandastore/mybackup/logs.

Задание 12. «Испортите» (переместите в /tmp) и восстановите файл /etc/hosts.

```
mv /etc/hosts /tmp/hosts.bak
```

```
cd /etc # Важно находиться в «правильном» каталоге
```

```
sudo -u backup amadmin mybackup find localhost /etc # найти ленту
```

Получим что-то похожее на строки ниже. Найдите строку с уровнем 0.

date	host	disk	lv	storage	pool	tape or file	file	part	status
2026-08-12	21:27:13	localhost	/etc	1	mybackup	mybackup	MYBACKUP-0005	2	1/1 OK
2026-08-12	21:54:41	localhost	/etc	1	mybackup	mybackup	MYBACKUP-0006	2	1/1 OK
2026-08-12	21:55:01	localhost	/etc	1	mybackup	mybackup	MYBACKUP-0001	2	1/1 OK
2026-08-12	21:55:25	localhost	/etc	1	mybackup	mybackup	MYBACKUP-0002	2	1/1 OK
2026-08-12	21:55:38	localhost	/etc	1	mybackup	mybackup	MYBACKUP-0003	2	1/1 OK
2026-08-12	21:56:44	localhost	/etc	0	mybackup	mybackup	MYBACKUP-0004	2	1/1 OK

Проверьте наличие файла hosts в резервной копии.

```
sudo -u backup amrestore -p
```

```
/amandastore/mybackup/vtapes/slot4/00002.localhost._etc.0 | tar -tf - ./hosts
```

Восстановите его

```
sudo -u backup amrestore -p
```

```
/amandastore/mybackup/vtapes/slot4/00002.localhost._etc.0 | tar -xf - ./hosts
```

Проверьте что файл восстановился

```
cat hosts
```

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 7. Язык сценариев Perl

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя root.

Глава 7. Язык сценариев Perl

7.1. Простые сценарии Perl

Задание 1. Напечатайте в отладчике 10 строк «Hello World».

Запустите сценарий под отладчиком:

```
perl -d -e 'for (1..10) { print "Hello World\n"; }'
```

В отладчике доступны команды: b <строка> — точка остановки, c — продолжить выполнение, n — шаг (не заходя в функции), s — шаг (с заходом в функции), p <выражение> — печать значения переменной.

Задание 2. Напишите сценарий, который создаёт массив чисел в диапазоне 9–0 и выводит его на экран.

Глава 7. Язык сценариев Perl

```
cat > arr.pl <<'EOF'
my @a = reverse 0..9;
print "@a\n";
EOF
perl arr.pl # 9 8 7 6 5 4 3 2 1 0
```

Задание 3. Создайте сценарий с ассоциативным массивом: ключи — имена, значения — возраст. Выведите значения элементов, их ключи и количество элементов.

```
cat > age.pl <<'EOF'
my %age = (Ivan=>30, Petr=>25, Anna=>28);
print "Возраст Ivan: $age{Ivan}\n";
print "Ключи: ", join(", ", keys %age), "\n";
print "Значения: ", join(", ", values %age), "\n";
print "Количество элементов: ", scalar(keys %age), "\n";
EOF
perl age.pl
```

Задание 4. Прервите вывод массива из предыдущего задания на третьем элементе.

```
cat > stop3.pl <<'EOF'
my %age = (Ivan=>30, Petr=>25, Anna=>28, Olga=>35);
my $i = 0;
for my $k (sort keys %age) {
    print "$k => $age{$k}\n";
    last if ++$i == 3;
}
EOF
perl stop3.pl
```

Оператор last прерывает цикл после третьей итерации.

Задание 5. Создайте четыре функции: сложение, вычитание, умножение и деление. Проверьте их работу.

```
cat > func.pl <<'EOF'
sub add { my ($x,$y)=@_; return $x+$y; }
sub sbt { my ($x,$y)=@_; return $x-$y; }
sub mul { my ($x,$y)=@_; return $x*$y; }
sub div { my ($x,$y)=@_; $y or die "Деление на ноль\n"; return $x/$y; }
print add(10,5), "\n";
print sbt(10,5), "\n";
print mul(10,5), "\n";
print div(10,5), "\n";
EOF
perl func.pl # 15 5 50 2
```

Задание 6. Измените сценарий так, чтобы аргументы и имя функции считывались из командной строки.

```
cat > calc.pl <<'EOF'
my ($op, $x, $y) = @ARGV;

sub add { my ($x,$y)=@_; return $x+$y; }
sub sbt { my ($x,$y)=@_; return $x-$y; }
sub mul { my ($x,$y)=@_; return $x*$y; }
sub div { my ($x,$y)=@_; $y or die "Деление на ноль\n"; return $x/$y; }

my %f = (add=>\&add, sub=>\&sbt, mul=>\&mul, div=>\&div);
die "Неизвестная операция: $op\n" unless exists $f{$op};
print $f{$op}->($x, $y), "\n";
EOF
perl calc.pl add 5 3 # 8
perl calc.pl div 10 2 # 5
```

Глава 7. Язык сценариев Perl

```
perl calc.pl mul 4 6 # 24
```

Задание 7. Создайте сценарий, который выводит «Hello имя_пользователя_запустившего_сценарий you login имя_компьютера_на_котором_запущен_сценарий».

```
cat > hello.pl <<'EOF'
use Sys::Hostname;
my $user = getpwuid($<) || $ENV{USER};
my $host = hostname() || $ENV{HOSTNAME};
print "Hello $user you login $host\n";
EOF
perl hello.pl
```

7.2. Файловые операции

Задание 1. Создайте ассоциативный массив: ключи — имена пользователей, значения — их UID. Выведите его на экран вместе с ключами.

```
cat > uids.pl <<'EOF'
my %uid;
while (my ($name, $passwd, $uid) = getpwent()) {
    $uid{$name} = $uid;
}
for my $user (sort keys %uid) {
    print "$user => $uid{$user}\n";
}
EOF
perl uids.pl
```

Задание 2. Создайте сценарий форматированного вывода содержимого текущего каталога. Имена файлов разделяются табуляцией.

```
cat > lsdir.pl <<'EOF'
opendir(my $dh, ".") or die $!;
my @files = readdir $dh;
closedir $dh;
print join("\t", sort @files), "\n";
EOF
perl lsdir.pl
```

Задание 3. Создайте сценарий, архивирующий содержимое вашего домашнего каталога в файл /tmp/день_недели.tar.bz2.

```
cat > backup.pl <<'EOF'
my @dow = qw(Воскресенье Понедельник Вторник Среда Четверг Пятница Суббота);
my $day = $dow[(localtime)[6]];
my $home = $ENV{HOME} || getpwuid($<);
my $file = "/tmp/$day.tar.bz2";
system("tar -cjf $file -C / " . substr($home, 1));
print "Создан архив $file\n";
EOF
perl backup.pl
```

Имя архива зависит от текущего дня недели, например /tmp/Вторник.tar.bz2.

Задание 4. Создайте сценарий, выводящий события из журнала только за дату, заданную аргументом.

```
cat > jour.pl <<'EOF'
my $date = shift @ARGV or die "Использование: $0 DATA\n";
open my $fh, "<", "/var/log/syslog" or die $!;
```

Глава 7. Язык сценариев Perl

```
while (<$fh>) { print if /^$date/; }
close $fh;
EOF
```

perl jour.pl "Aug 10" # дата в формате журнала (например «авг 10» в русской локали)

Задание 5. Создайте файл с содержимым: ip=127.0.0.1 some string abc; ip=1.2.3.4 some string 123. Создайте сценарий, заменяющий IP 1.2.3.4 в этом файле на 10.1.2.3.

```
cat > ips.txt <<'EOF'
ip=127.0.0.1 some string abc
ip=1.2.3.4 some string 123
EOF
```

Однострочная замена:

```
perl -pi -e 's/ip=1\.2\.3\.4/ip=10.1.2.3/g' ips.txt
cat ips.txt
```

Вариант отдельным сценарием:

```
cat > fix.pl <<'EOF'
open my $in, "<", "ips.txt" or die $!;
my $out = join("", <$in>);
$out =~ s/ip=1\.2\.3\.4/ip=10.1.2.3/g;
print $out;
EOF
perl fix.pl
```

Задание 6. Создайте сценарий, формирующий HTML-страницу со списком пользователей, работающих в системе.

```
cat > users.pl <<'EOF'
use strict;
use warnings;
print "Content-Type: text/html; charset=utf-8\n\n";
print "<html><body><h1>Пользователи в системе</h1><ul>\n";
open my $fh, "-|", "who" or die $!;
while (<$fh>) {
    my ($user) = split /\s+/;
    print "<li>$user</li>\n";
}
close $fh;
print "</ul></body></html>\n";
EOF
perl users.pl
```

Задание 7. Проверьте работу сценария CGI в taint-режиме.

Запуск сценария с флагом -T (или первая строка #!/usr/bin/perl -T) включает taint-режим: все данные из внешнего мира (окружение, аргументы командной строки, ввод) считаются небезопасными.

```
chmod +x users.pl
perl -T users.pl
```

При использовании небезопасных данных в чувствительных операциях возникнет ошибка «Insecure dependency ... while running with -T switch». Чтобы снять «порчу» (untaint), значение проверяется регулярным выражением, а подставленная группа берётся в новых скобках:

```
my ($user) = $user =~ /^([a-z0-9_]+)$/;
```

или самостоятельно определить переменную:

```
$ENV{PATH}='/usr/local/sbin:/usr/local/bin:/sbin:/bin';
```

Разместите сценарий в каталоге CGI (например /var/www/cgi-bin) и проверьте в браузере или

curl.

L-201. Лабораторные работы. Ответы (версия 1.3)

Глава 8. Решение проблем

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя root.

8.1. Решение проблем

Задание 1. Запустите систему без /sbin/init и поменяйте пароль пользователю root.

1. В меню GRUB выберите нужный пункт меню и нажмите e.
2. В строке, начинающейся с linux (или linux16), допишите параметр: init=/bin/bash
3. Нажмите Ctrl+X — система загрузится в оболочку bash без загрузчика init.
4. Перемонтируйте корневую ФС в режиме чтения/записи:

```
mount -o remount,rw /
```

5. Смените пароль root:

```
passwd root
```

6. Если используется SELinux, создайте пустой файл для пересоздания меток ФС:

```
touch /.autorelabel
```

7. Выйдите из оболочки и перезагрузите систему:

```
exec /sbin/init
```

```
reboot
```

8. Проверьте вход под root с новым паролем.

Задание 2. Проверьте, нормально ли работает ОС.

```
uptime
```

```
systemctl --failed
```

```
journalctl -p err -b
```

```
dmesg | grep -iE "error|fail"
```

Задание 3. Если имеются проблемы с какими-то службами, выясните, что не работает и почему.

```
systemctl list-units --failed
```

```
systemctl status <имя_службы>
```

```
journalctl -u <имя_службы> -n 50
```

```
tail -n 50 /var/log/syslog
```

Задание 4. Посмотрите, какие журналы загрузки у вас имеются.

В Debian используется journald; журналы загрузки хранятся в systemd-journald:

```
journalctl -b # журнал текущей загрузки
```

```
journalctl --list-boots # список всех загрузок
```

```
ls -l /var/log/ # также /var/log/syslog (если включён rsyslog)
```

Задание 5. Проанализируйте журнал предыдущей загрузки. Были ли там проблемы.

```
journalctl -b -1
```

```
journalctl -b -1 -p err # только ошибки предыдущей загрузки
```

Задание 6. Установите цель загрузки по умолчанию basic.target.

Глава 8. Решение проблем

```
systemctl set-default basic.target
systemctl get-default # basic.target
```

Задание 7. Перезапустите машину. Можете ли вы взаимодействовать с ОС?

```
reboot
```

В `basic.target` не запускаются `getty` и большинство служб многопользовательского режима, поэтому консольного входа в систему нет. Система в минимальном состоянии; при необходимости верните цель `multi-user.target` (это можно сделать параметром ядра `systemd.unit=multi-user.target`).

Задание 8. Загрузите компьютер, установив цель загрузки `multi-user.target` через параметр ядра.

В меню GRUB выберите пункт меню и нажмите `e`. В строке `linux` допишите параметр:

```
systemd.unit=multi-user.target
```

Нажмите `Ctrl+X` и дождитесь загрузки. Проверка:

```
systemctl get-default
```

Задание 9. Установите цель загрузки по умолчанию `multi-user.target`.

```
systemctl set-default multi-user.target
```

Задание 10. Перезапустите машину и убедитесь в корректной работе ОС.

```
reboot
```

```
systemctl get-default # multi-user.target
```

```
systemctl --failed # без ошибок
```

Задание 11. Выясните, какие системные вызовы производит команда `ls`.

```
apt-get install -y strace
```

```
strace -o /tmp/ls.strace ls
```

```
grep -E "execve|open(at)?\(|fstat|getdents" /tmp/ls.strace | head
```

Пример вызова трассировки: `openat(AT_FDCWD, ".", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC)`.

Задание 12. Определите, какие библиотечные функции вызываются командой `ls`.

```
apt-get install -y ltrace
```

```
ltrace -o /tmp/ls.ltrace ls
```

```
grep -E "malloc|strcmp|strlen" /tmp/ls.ltrace | head
```

Задание 13. Настройте вашу систему на использование русского языка.

Убедитесь, что локаль добавлена в `/etc/locale.gen`, и выполните генерацию:

```
grep ru_RU /etc/locale.gen
```

```
locale-gen ru_RU.UTF-8
```

Установите локаль по умолчанию:

```
update-locale LANG=ru_RU.UTF-8
```

или с помощью команды:

```
localectl set-locale LANG=ru_RU.UTF-8
```

```
locale && env | grep LANG
```

Задание 14. Проверьте, какие локали установлены.

```
locale -a
```

```
localectl list-locales
```

Задание 15. Установите локаль для китайского языка и проверьте.

Добавьте строку в `/etc/locale.gen` или выполните напрямую:

```
locale-gen zh_CN.UTF-8
```

```
locale -a | grep zh # например zh_CN.UTF-8, zh_CN.GB18030
```

Задание 16. Разрешите пользователю `user1` выполнять команду `passwd` через утилиту `sudo`.

```
apt-get install -y sudo  
visudo
```

Добавьте строку:

```
user1 ALL=(ALL) /usr/bin/passwd
```

Проверка:

```
sudo -l -U user1  
su - user1 -c 'sudo passwd'
```

L-201. Лабораторные работы. Ответы (версия 1.3)

ОС назначения — Debian GNU/Linux. Команды выполняются от имени пользователя `root`.

Глава 9. Инструменты поддержки версий

9.1. RCS

Задание 1. Создайте ревизию файла `/etc/hosts`.

Для лабораторной удобнее работать с копией в отдельном каталоге (RCS создаёт историю в подкаталоге RCS):

```
mkdir -p /tmp/rcs && cd /tmp/rcs && cp /etc/hosts .  
apt-get install -y rcs  
ci -l hosts
```

Команда `ci` создаёт файл `RCS/hosts,v` с ревизией 1.1. Флаг `-l` оставляет файл с блокировкой (доступным для дальнейшего редактирования).

Задание 2. Сделайте файл доступным для редактирования, внесите изменения и добавьте макрос `$Id$`.

После `ci -l` файл остаётся доступным для правки. Добавьте строку с макросом:

```
echo '# $Id$' >> hosts
```

При выкачивании (`co`) макрос раскрывается автоматически:

```
co -l hosts  
grep Id hosts # видна строка: # $Id: hosts,v 1.1 ... $
```

Задание 3. Создайте вторую ревизию файла и получите список изменений в ревизиях.

```
ci -l hosts # создаётся ревизия 1.2  
rlog hosts # лог всех ревизий с комментариями  
rcsdiff -r1.1 -r1.2 hosts # различия между ревизиями
```

9.2. CVS

Задание 1. Создайте репозиторий в каталоге `/home/cvs`.

```
apt-get install -y cvs  
cvs -d /home/cvs init
```

Будет создана служебная структура `/home/cvs/CVSROOT`.

Глава 9. Инструменты поддержки версий

Задание 2. Настройте переменную CVSROOT на использование этого каталога.

```
echo 'export CVSROOT=/home/cvs' >> ~/.bashrc
source ~/.bashrc
echo $CVSROOT
```

Задание 3. Поместите в репозиторий каталог /etc.

```
cd /etc
cvs import -m "Initial import of /etc" etc vendor start
```

Команда `cvs import` создаёт в репозитории модуль `etc` с исходным состоянием каталога.

Задание 4. Измените файл /etc/hosts.allow, добавив к нему макросы \$Id\$ и \$Log\$.

Возьмите рабочую копию:

```
cd /tmp
cvs checkout etc
cd etc
echo '# $Id$' >> hosts.allow
echo '# $Log$' >> hosts.allow
```

Задание 5. Подтвердите изменения и получите список изменений в первой и текущей версии файла.

```
cvs commit -m "Добавлены макросы Id и Log" hosts.allow
cvs log hosts.allow
cvs diff -u -r1.1 -r1.2 hosts.allow # первая и текущая ревизии
```