

Оглавление

Лабораторная работа 1. Установка ОС Альт.....	2
Предварительно.....	2
Задание 1.1. Установка ОС Альт Сервер.....	2
Задание 1.2. Установка ОС Альт Рабочая станция.....	2
Лабораторная работа 2. Запуск и останов системы и ПО.....	2
Задание 2.1. Загрузка в текстовой режим (третий уровень исполнения).....	2
Задание 2.2. Загрузка с замененным процессом init.....	3
Задание 2.3. Создание описания для сервиса.....	3
Задание 2.4. Запуск-статус-остановка сервиса.....	4
Задание 2.5. Управление сервисами.....	4
Задание 2.6. Журнал событий journald.....	5
Задание 2.7. Запросы средствами journalctl.....	5
Дополнительная лабораторная. Работа с дисковой подсистемой.....	6
Предварительная информация.....	6
Задание 3.1. Разбиение дисков средствами fdisk и parted.....	6
Задание 3.2. Управление томами LVM.....	8
Задание 3.3. Создание программного RAID средствами mdadm.....	10
Лабораторная работа 4. Мониторинг работы системы.....	13
Задание 4.1. Установка и знакомство с утилитой stress.....	13
Задание 4.2. Создание нагрузки по процессору.....	13
Задание 4.3. Создание нагрузки по памяти.....	14
Задание 4.4. Отключение подкачки страниц (свопинга).....	14
Задание 4.5. Создание сверх-нагрузки по памяти.....	15

Лабораторная работа 1. Установка ОС Альт

Предварительно

1. Скачайте дистрибутивы для архитектуры x86_64, платформа Р11
 - Альт Сервер
 - Альт Рабочая станция

```
https://getalt.org
```

Задание 1.1. Установка ОС Альт Сервер

1. Создайте в среде виртуализации (VirtualBox или другой используемой) виртуальную машину с характеристиками, соответствующими требованиям ОС Альт Сервер
 - Подсоедините к ВМ ISO-образ с дистрибутивом Альт Сервера
2. Выполните установку ОС Альт Сервер в рамках созданной виртуальной машины.
 - используйте параметры по умолчанию
 - установите графический интерфейс (GNOME)

Задание 1.2. Установка ОС Альт Рабочая станция

1. Создайте в среде виртуализации (VirtualBox или другой используемой) виртуальную машину с характеристиками, соответствующими требованиям ОС Альт Рабочая станция
 - Подсоедините к ВМ ISO-образ с дистрибутивом Альт Рабочая станция
2. Выполните установку ОС Альт Рабочая станция в рамках созданной виртуальной машины.
 - используйте параметры по умолчанию

Лабораторная работа 2. Запуск и останов системы и ПО

Задание 2.1. Загрузка в текстовой режим (третий уровень исполнения)

- Задание выполняется в контексте безопасности суперпользователя
1. Посмотрите текущий уровень работы операционной системы инструментарием SystemV

```
# runlevel
```

2. Посмотрите цель инициализации системы по-умолчанию

```
# systemctl get-default
```

3. Выполните перезагрузку системы
4. В момент работы загрузчика **GRUB** перейдите в режим редактирования опций загрузки
 - **e**
5. Допишите в строчке загрузки ядра - начинается с **linux** указание загрузиться в Зий уровень выполнения, дописав в конце строки номер уровня
 - **3**
6. Выполните загрузку
 - **F10**
7. По окончании загрузки войдите в систему и перейдите в контекст безопасности суперпользователя
8. Посмотрите текущий уровень работы операционной системы инструментарием SystemV

```
# runlevel
```

Задание 2.2. Загрузка с замененным процессом init

- Задание выполняется в контексте безопасности суперпользователя
1. С использованием команды **which** определите путь к исполняемому файлу командного интерпретатора **bash**. Запомните/запишите его.
 2. Выполните перезагрузку системы
 3. В момент работы загрузчика **GRUB** перейдите в режим редактирования опций загрузки
 4. Допишите в строчке загрузке ядра ОС необходимость использовать командный интерпретатор вместо процесса **init**
 - **init=<путь к bash>**
 5. Выполните загрузку операционной системы

Задание 2.3. Создание описания для сервиса

1. Перезагрузите систему в режиме загрузки по-умолчанию
2. Задание выполняется в контексте безопасности суперпользователя
3. Запросите с **github.com** получение дополнительных файлов для проведения лабораторных занятий

```
$ git clone https://github.com/hse-labs/linux-lf.git
```

4. При необходимости выполните установку пакета **git-core** средствами **apt-get**

```
# apt-get install git-core
```

5. Ознакомьтесь с содержимым файла **fake.service** (Unit-файл сервиса) в полученном каталоге

```
# cat linux-lf/fake.service
[Unit]
Description=fake
After=network.target
[Service]
ExecStart=/bin/sh -c '/bin/echo "I am starting the fake service" ; /bin/sleep 30'
ExecStop=/bin/echo "I am stopping the fake service"
[Install]
WantedBy=multi-user.target
```

6. Ознакомьтесь с текущим списком описаний сервисов для SystemD

```
# ls /etc/systemd/system
# ls /lib/systemd/system
```

6. Скопируйте файл **fake.service** с описанием сервиса в каталог **/etc/systemd/system**

```
# cp linux-lf/fake.service /etc/systemd/system/
```

7. Убедитесь, что описание сервиса скопировано

```
# ls /etc/systemd/system/fake.service
```

Задание 2.4. Запуск-статус-остановка сервиса

1. Задание выполняется в контексте безопасности суперпользователя
 - Для выполнения дальнейших действий используйте утилиту **systemctl**
2. Выполните запуск созданного сервиса **fake.service**
3. Посмотрите статус работы сервиса
4. Перезапустите созданный сервис
5. Остановите сервис
6. Настройте сервис на автоматический запуск

Задание 2.5. Управление сервисами

1. Задание выполняется в контексте безопасности суперпользователя
 - Для выполнения дальнейших действий (п.2,4,6,7,8,10) используйте утилиту **systemctl**
2. Уберите сервис CUPS (**cups.service**) из автозагрузки.
3. Перезагрузитесь в третий уровень загрузки.

```
# telinit 3
```

4. Проверьте, запущен ли CUPS.

5. Залогиньтесь пользователем sysadmin и перейдите в пятый уровень загрузки.

```
# telinit 5
```

6. Проверьте, запущен ли CUPS.
7. Запустите CUPS.
8. Верните CUPS в автозагрузку.
9. Перезагрузитесь в пятый уровень загрузки

```
# telinit 5
```

10. Проверьте, запущен ли CUPS.

Задание 2.6. Журнал событий **journald**

1. Задание выполняется в контексте безопасности суперпользователя
2. Ознакомьтесь с настройками системы журнализации **journald**

```
# cat /etc/systemd/journald.conf
```

3. Обратите внимание на значение параметров
 - **Storage**
 - **ForwardToSyslog**
4. Ознакомьтесь с расположением файлов журналов **journald**
 - в каталоге на диске **/var/log/journal** (если есть)
 - в каталоге в RAM-диске **/run/log/journal** (если есть)

Задание 2.7. Запросы средствами **journalctl**

1. Задание выполняется в контексте безопасности суперпользователя
2. Ознакомьтесь со встроенной справкой утилиты **journalctl**

```
$ man journalctl  
$ journalctl --help
```

3. Выполните вывод событий приоритета **emerg**

```
# journalctl -p emerg
```

4. Выполните вывод событий ядра

```
# journalctl -k
```

5. Выполните вывод событий службы **sshd**

```
# journalctl $(which sshd)
```

6. Выполните вывод событий пользователя **sysadmin**

```
# journalctl _UID=$(id -u sysadmin)
```

7. Выполните вывод событий с момента загрузки системы

```
# journalctl -b
```

8. Выполните вывод последних сообщений с возможной расшифровкой

```
# journalctl -xe
```

Дополнительная лабораторная. Работа с дисковой подсистемой

Предварительная информация

- Для выполнения заданий предварительно предполагается добавление к виртуальным машинам дополнительных накопителей (/dev/sdb, /dev/sdc). Выполняется средствами используемой системы виртуализации.
- Перед выполнением заданий рекомендуется создать снимок с состоянием используемых виртуальных машин. Далее в заданиях упоминается как снимок **Старт**

Задание 3.1. Разбиение дисков средствами **fdisk** и **parted**

Информация по блочным устройствам

- Ознакомьтесь со встроенной справкой по утилите **lsblk**

```
$ man lsblk  
$ lsblk --help
```

- Посмотрите информацию по блочным устройствам системы
 - обратите внимание на диски системы: **TYPE=disk**
 - обратите внимание на имеющиеся дисковые разделы: **TYPE=part**

```
$ lsblk
```

- Ознакомьтесь со встроенной справкой по утилите **blkid**

```
$ man blkid  
$ blkid --help
```

- Посмотрите информацию по разделу с системой

```
$ blkid /dev/sda1
```

Использование fdisk

- Задание выполняется в контексте безопасности суперпользователя
- Запустите **fdisk** для диска **/dev/sdb**

```
$ fdisk /dev/sdb
```

- Посмотрите сводку по командам **fdisk**
 - команда **m**
- Посмотрите текущую таблицу разделов
 - команда **p**
- По выводу таблицы определите имеющиеся на диске разделы (если есть) и определите тип таблицы разделов - **mbr** (обозначена как dos) или **gpt**
- Создайте новый первичный раздел размером 1Гб
 - команда **n**
 - укажите тип - первичный - команда **p** (не требуется для gpt-дисков)
 - укажите порядковый номер (по умолчанию)
 - укажите первый сектор (по умолчанию)
 - укажите размер **+1G**
- Посмотрите получившуюся таблицу разделов
 - команда **p**
- Запишите таблицу разделов на диск и выйдите из **fdisk**
 - команда **w**
- Убедитесь, что в системе появилось новое дисковое устройство

```
$ ls -l /dev/sd*
$ lsblk
```

Работа с созданным разделом

- Задание выполняется в контексте безопасности суперпользователя
- Убедитесь в доступности созданного на **/dev/sdb** раздела средствами **lsblk**

```
$ _____
```

- Создайте на разделе файловую систему **ext4**

```
$ _____
```

- Создайте подкаталог в каталоге **/mnt** для монтирования и примонтируйте созданную ФС в него

```
$ _____
```

- Проверьте возможность работы с примонтированной ФС - создайте там файл, запишите туда какие-либо данные, прочтайте

```
$ _____
```

- Выполните отмонтирование раздела

```
$ _____
```

Задание 3.2. Управление томами LVM

Подготовка гипервизора

- Выполните восстановление используемой Вами ВМ до снимка с именем **Старт**

Создание разделов для LVM

- Задание выполняется в контексте безопасности суперпользователя
- Средствами **fdisk** создайте 2 раздела по 500Мб на разных физических дисках **sdb** и **sdc**
 - При создании укажите тип раздела **p** - primary (не требуется для gpt-дисков)

```
$ _____
```

- Промаркируйте раздел (команда **t**) как **8e** - Linux LVM (не требуется для gpt-дисков)
- При помощи **lsblk** убедитесь, что разделы созданы и имеют одинаковый размер

```
$ _____
```

Создание PV

- Задание выполняется в контексте безопасности суперпользователя
- Создайте PV из подготовленных разделов

```
$ pvcreate /dev/sdbX  
$ pvcreate /dev/sdcY
```

- Где X, Y - правильные номера разделов (ранее созданные)
- Убедитесь в корректности создания PV

```
$ pvdisk
```

Создание VG

- Задание выполняется в контексте безопасности суперпользователя

- Создайте VG менем **vg01** из подготовленных PV

```
$ vgcreate vg01 /dev/sdbX /dev/sdcY
```

- Где X, Y - правильные номера разделов (ранее созданные)
- Убедитесь в корректности создания VG

```
$ vgdisk
```

Создание LV

- Задание выполняется в контексте безопасности суперпользователя
- Создайте LV размером 400Мб и именем **lv01** из пространства, доступного в VG **vg01**

```
$ lvcreate -L 400M -n lv01 vg01
```

- Убедитесь в корректности создания LV

```
$ lvscan  
$ lvdisk  
$ lvs
```

- Убедитесь в появлении в системе нового блочного устройства

```
$ lsblk
```

Создание ФС, монтирование

- Задание выполняется в контексте безопасности суперпользователя
- Создайте на LV-устройстве **/dev/vg01/lv01** ФС **ext4**

```
$ _____
```

- Создайте каталог **/mnt/lvmmnt**

```
$ _____
```

- Примонтируйте созданную ФС

```
$ _____
```

- Проверьте возможность работы с примонтированной ФС

```
$ _____
```

- Добавьте в **/etc/fstab** запись для автоматического монтирования LVM-раздела используя UUID раздела

```
$ _____
```

- Перезагрузите ВМ и убедитесь, что LVM-раздел подмонтировался

```
$ _____
```

Изменение размера раздела

- Задание выполняется в контексте безопасности суперпользователя
- При помощи утилиты **lvresize** увеличьте размер раздела на 100Мб

```
$ _____
```

- Убедитесь средствами **df**, **lsblk**, **lvdisplay** что размер увеличен

```
$ _____
```

Разбор LVM

- Самостоятельно выполните удаление созданных LV, VG, PV

```
$ _____  
$ _____  
$ _____
```

- Удалите из **/etc/fstab** созданные ранее записи монтирования разделов LVM

```
$ _____
```

Задание 3.3. Создание программного RAID средствами mdadm

Подготовка гипервизора

- Выполните восстановление используемой Вами ВМ до снимка с именем **Старт**

Создание разделов для RAID

- Задание выполняется в контексте безопасности суперпользователя
- Средствами **fdisk** создайте 2 раздела по 500Мб на разных физических дисках **sdb** и **sdc**
 - Укажите тип раздела **p** - primary (не требуется для gpt-дисков)

```
$ _____
```

- Промаркируйте раздел (команда **t**) как **fd** - Linux Raid Autodetect (не требуется для gpt-дисков)
- При помощи **lsblk** убедитесь, что разделы созданы и имеют одинаковый размер

```
$ _____
```

Создание программного RAID

- Задание выполняется в контексте безопасности суперпользователя
- Создайте из двух разделов, созданных ранее, устройство **RAID1** с именем **/dev/md0**

```
$ mdadm -C /dev/md0 -l 1 -n 2 /dev/sdbX /dev/sdcY
```

- Где X, Y - правильные номера разделов (ранее созданные)
- При запросе подтверждения, подтвердите создание массива - **y**
- Убедитесь в том, что **RAID** создан и работоспособен

```
$ cat /proc/mdstat  
$ mdadm -D /dev/md0
```

- Обратите внимание на статусы дисков в массиве **[UU]**
- Посмотрите метаданные созданного **RAID-массива**

```
$ mdadm --detail --scan
```

- Запишите метаданные созданного **RAID-массива** в **mdadm.conf**

```
$ mdadm --detail --scan >> /etc/mdadm.conf
```

- Обновите загрузочный образ файловой системы

```
$ make-initrd
```

- По выводу команды убедитесь, что загрузочный образ файловой системы создан

Создание ФС, монтирование

- Задание выполняется в контексте безопасности суперпользователя
- Создайте на RAID-устройстве **/dev/md0** ФС **ext4**

```
$ _____
```

- Создайте каталог **/mnt/raidmnt**

```
$ _____
```

- Примонтируйте созданную ФС

```
$ _____
```

- Проверьте возможность работы с примонтированной ФС

```
$ _____
```

- Добавьте в **/etc/fstab** запись для автоматического монтирования RAID-раздела

```
$ _____
```

- Перезагрузите ВМ и убедитесь, что RAID-массив подмонтировался

```
$ _____
```

Замена сбояного диска в массиве

- Задание выполняется в контексте безопасности суперпользователя
- Средствами **fdisk** создайте раздел размером 500Мб на физическом диске **sdb**
 - Укажите тип раздела **fd**

```
$ _____
```

- При помощи **lsblk** убедитесь, что раздел создан

```
$ _____
```

- В созданном ранее RAID-массиве пометьте один из дисков массива как сбойный

```
$ _____
```

- Средствами диагностики состояния массива убедитесь, что он “развалился”

```
$ _____
```

- Добавьте к массиву ранее созданный раздел

```
$ _____
```

- Средствами диагностики проследите процесс восстановления массива и убедитесь что массив восстановился

```
$ _____
```

- Убедитесь в наличии на массиве ранее созданных на нем данных

```
$ _____
```

Создание программного RAID5

- Выполните восстановление используемой Вами ВМ до снимка с именем **Старт**
- Задание выполняется в контексте безопасности суперпользователя.
- Создайте на дисках /dev/sdb и /dev/sdc 3 раздела размером по 1G.

- Объедините созданные разделы в RAID5.
- Убедитесь, что массив успешно создан.
- Создайте на массиве файловую систему ext4.
- Обеспечьте доступность созданной файловой системы в каталоге /mnt/raid5 (создайте каталог при необходимости).
- Добавьте запись в /etc/fstab для автоматического монтирования данного раздела в указанный каталог при загрузке системы.
- Сохраните конфигурацию массива в файле **mdadm.conf**.
- Обновите загрузочный образ файловой системы.
- Перезагрузитесь и убедитесь в доступности массива после перезагрузки.

Лабораторная работа 4. Мониторинг работы системы

Задание 4.1. Установка и знакомство с утилитой **stress**

- Задание выполняется в контексте безопасности суперпользователя
1. Поставьте утилиту **stress** средствами менеджера зависимостей
 - Утилита предназначена для создания нагрузки на систему. В следующих заданиях мы будем использовать ее для эмуляции стрессовой нагрузки и отслеживания данной нагрузки инструментами мониторинга.

```
# apt-get install stress
```

2. Ознакомьтесь со встроенной справкой по его использованию

```
$ man stress
$ stress --help
```

Задание 4.2. Создание нагрузки по процессору

- Задание выполняется в контексте безопасности суперпользователя
1. Продолжайте выполнять задание в той же ВМ, в которой делали предыдущее
 2. При помощи утилиты **lscpu** определите количество доступных процессорных ядер в вашей системе

```
# lscpu
```

2. Запустите утилиту **stress** с опциями для создания 2 процессов, каждый из которых интенсивно (полностью) использует CPU на 30 секунд

```
# stress -c 2 -t 30s &
```

3. Средствами мониторинга **top** оцените создаваемую нагрузку

```
# top
```

4. Сколько процессов **stress** вы видите, какую нагрузку на процессор создает каждый из них?
5. Как бы выглядела загрузка CPU при данном вызове **stress** для случая системы с 1 ядром CPU, 2 ядер, 4 ядер?
6. Дождавшись завершения 30-секундного интервала создаваемой нагрузки убедитесь, что **stress** больше не потребляет ресурсов процессора
7. Завершите работу с утилитой **top** при помощи комбинации **Ctrl-C**

Задание 4.3. Создание нагрузки по памяти

- Задание выполняется в контексте безопасности суперпользователя

 1. Продолжайте выполнять задание в той же ВМ, в которой делали предыдущее
 2. При помощи утилиты **free** определите текущее распределение памяти в вашей системе

```
# free -m
```

3. Запустите утилиту **stress** с опциями для создания 2 процессов интенсивно использующих память на 30 секунд

```
# stress -m 2 -t 30s &
```

4. При помощи утилиты **free** определите текущее распределение памяти в вашей системе

```
# free -m
```

Задание 4.4. Отключение подкачки страниц (свопинга)

- Задание выполняется в контексте безопасности суперпользователя

 1. Продолжайте выполнять задание в той же ВМ, в которой делали предыдущее
 2. Посмотрите текущие используемые своп-разделы

```
$ cat /proc/swaps
```

3. Отключите механизм подкачки страниц в системе

```
# swapoff -a
```

4. Посмотрите текущие используемые своп-разделы

```
# cat /proc/swaps
```

5. Убедитесь, что система более не использует механизм подкачки страниц

Задание 4.5. Создание сверх-нагрузки по памяти

1. Продолжайте работу в той же ВМ, в которой вы ранее отключили свопинг
 - Задание выполняется в контексте безопасности суперпользователя
2. Запустите утилиту **stress** опциями для создания 8 процессов интенсивно использующих память на 30 секунд

```
# stress -m 8 -t 30s
```

3. Вы должны увидеть действие **OOM (Out of Memory) killer**
4. Проследите действие ООМ killer по **dmesg** и журналам

```
# dmesg  
# journalctl -xe
```

5. Обратите внимание на сообщения об исчерпании памяти и действию ООМ по завершению процессов, потребляющих память