

Управление загрузкой системы

Отчет

Казначеев Сергей Ильич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	12
5	Выводы	13

Список иллюстраций

3.1	1	7
3.2	2	7
3.3	2_2	7
3.4	3	8
3.5	4	8
3.6	5	8
3.7	6	9
3.8	7	9
3.9	8	9
3.10	9	9
3.11	10	10
3.12	11	10
3.13	12	10
3.14	13	11
3.15	14	11
3.16	15	11

Список таблиц

1 Цель работы

Получить навыки работы с загрузчиком системы GRUB2.

2 Задание

1. Продемонстрируйте навыки по изменению параметров GRUB и записи изменений в файл конфигурации
2. Продемонстрируйте навыки устранения неполадок при работе с GRUB
3. Продемонстрируйте навыки работы с GRUB без использования root

3 Выполнение лабораторной работы

Перейдем в супер пользователя

```
[sikaznacheev@localhost ~]$ su -  
Пароль:  
[root@localhost ~]#
```

Рис. 3.1: 1

После чего в файле /etc/default/grub установи параметр отображения меню загрузки в течение 10 секунд

```
GRUB_TIMEOUT=10  
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"  
GRUB_DEFAULT=saved  
GRUB_DISABLE_SUBMENU=true  
GRUB_TERMINAL_OUTPUT="console"  
GRUB_CMDLINE_LINUX="crashkernel=16-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/r1-swap rd.lvm.lv=r1/root rd.lvm.lv=r1/sw"   
GRUB_DISABLE_RECOVERY="true"  
GRUB_ENABLE_BLSCFG=true
```

Рис. 3.2: 2

Просмотр изний

```
Use the ↑ and ↓ keys to select which entry is highlighted.  
Press enter to boot the selected OS, 'e' to edit the commands  
before booting or 'c' for a command-line.  
The highlighted entry will be executed automatically in 7s.
```

Рис. 3.3: 2_2

После сохранения файл и закрываем редактор и записываем изменения в GRUB2 введя `grub2-mkconfig > /boot/grub2/grub.cfg`

```
[root@localhost ~]# grub2-mkconfig > /boot/grub2/grub.cfg
Generating grub configuration file ...
Adding boot menu entry for UEFI Firmware Settings ...
done
```

Рис. 3.4: 3

Затем перезагружаем систему, как только появляется GRUB выбираем строку текущей версии ядра в меню и нажимаем е для редактирования добавляем в конце строки systemd.unit=rescue.target

```
load_video
set gfxpayload=keep
insmod gzio
linux ($root)/vmlinuz-0-rescue-24895b72da964397829f89f50c792b1b root=/dev/m\
apper/rl-root ro crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/m\
apper/rl-swap rd.lvm.lv=rl/root rd.lvm.lv=rl/swap systemd.unit=rescue.targe\
t_
initrd ($root)/initramfs-0-rescue-24895b72da964397829f89f50c792b1b.img
```

Рис. 3.5: 4

Далее посмотрим список всех файлов модулей, которые загружены в настоящее время `systemctl list-units`

```
[root@localhost ~]# systemctl list-units_
```

Рис. 3.6: 5

Сам вывод

rescue.service	loaded	active	excited	Initiating rd.crypt services, components etc.
nfs-downname.service	loaded	active	excited	Read and set NFS downname from /etc/sys
plymouth-read-sectle.service	loaded	active	excited	Tell Plymouth To Write Out Runtime Data
plymouth-start.service	loaded	active	excited	Show Plymouth Root Screen
rescue.service	loaded	active	running	Rescue Shell
systemd-boot-update.service	loaded	active	excited	Automatic Boot Loader Update
systemd-journal-flush.service	loaded	active	excited	Flush Journal to Persistent Storage
systemd-journald.service	loaded	active	running	Journal Service
system-modules-load.service	loaded	active	excited	Load Kernel Modules
system-network-generator.service	loaded	active	excited	Generate network units from Kernel command
system-random-seed.service	loaded	active	excited	Load/Save OS Random Seed
system-resmount-fs.service	loaded	active	excited	Remount Root and Kernel File Systems
system-resolved.service	loaded	active	running	Network Name Resolution
system-sysctl.service	loaded	active	excited	Apply Kernel Variables
system-tagfiles-setup-dev.service	loaded	active	excited	Create Static Device Nodes in /dev
system-tagfiles-setup.service	loaded	active	excited	Create Udev Files and Directories
system-udev-settle.service	loaded	active	excited	Wait for udev To Complete Device Initializ
system-udev-trigger.service	loaded	active	excited	Collating all udev Devices
system-udev.service	loaded	active	running	Rule-based Manager for Device Events and P
system-update-utmp.service	loaded	active	excited	Record System Boot/Shutdown in UTMP
system-ocmole-setup.service	loaded	active	active	Setup Virtual Console
-.slice	loaded	active	active	Root Slice
system-mdprobe.slice	loaded	active	active	Slice /system/mdprobe
system-systemd-hibernate-v2resume.slice	loaded	active	active	Slice /system/systemd-hibernate-resume
system.slice	loaded	active	active	System Slice
dm-event.socket	loaded	active	listening	Device-mapper event daemon FIFOs
lvm2-lvpoll.socket	loaded	active	listening	LVM poll daemon socket
systemd-journald-dev-log.socket	loaded	active	running	Journal Socket (/dev/log)
systemd-journald.socket	loaded	active	running	Journal Socket
systemd-udev-control.socket	loaded	active	running	udev Control Socket
systemd-udev-kernel.socket	loaded	active	running	udev Kernel Socket
dev-mapper-r1-v2dmmap.map	loaded	active	active	/dev/mapper/r1-map
cryptsetup.target	loaded	active	active	Local Encrypted Unibus
integritysetup.target	loaded	active	active	Local Integrity Protected Unibus
local-fs-pre.target	loaded	active	active	Preparation for Local File Systems
local-fs.target	loaded	active	active	Local File Systems
network-pre.target	loaded	active	active	Preparation for Network
nss-lookup.target	loaded	active	active	Host and Network Name Lookups
rescue.target	loaded	active	active	Rescue Mode
sound.target	loaded	active	active	Sound Card
snmp.target	loaded	active	active	Snmps
sysinit.target	loaded	active	active	System Initialization
veritysetup.target	loaded	active	active	Local Verity Protected Unibus

Рис. 3.7: 6

После чего просмотрим задействованные переменные среды оболочки
systemctl show-environment

```
[root@localhost ~]# systemctl show-environment
LANG=ru_RU.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
[root@localhost ~]#
```

Рис. 3.8: 7

Затем перезагрузим систему

```
[root@localhost ~]# systemctl reboot
```

Рис. 3.9: 8

После того как отображается меню GRUB выбираем в меню строку версии
ядра системы и нажимаем е чтобы отредактировать и добавляем в конце
systemd.unit=emergency.target

```
set gfxpayload=keep
insmod gzio
linux ($root)/vmlinuz-5.14.0-570.55.1.el9_6.x86_64 root=/dev/mapper/r1-root\
ro crashkernel=16-46:192M,46-646:256M,646-:512M resume=/dev/mapper/r1-swap\
rd.lvm.lv=r1/root rd.lvm.lv=r1/swap systemd.unit=emergency.target
initrd ($root)/initramfs-5.14.0-570.55.1.el9_6.x86_64.img $tuned_initrd
```

Рис. 3.10: 9

Снова просматриваем список всех файлов модулей, которые загружены в настоящее время так же посмотрим задействованные переменные среды оболочки и перезапустим систему

[illegible]

Рис. 3.11: 10

После чего переходим в меню строку версии ядра системы и нажимаем е чтобы отредактировать и добавляем в конце `rd.break`

```
load_video
set gfxpayload=keep
insmod gzio
linux ($root)/vmlinuz-5.14.0-570.55.1.el9_6.x86_64 root=/dev/mapper/rl-root\
ro crashkernel=1G-4G:192M,4G-64G:256M,64G-512M resume=/dev/mapper/rl-swap\
rd.lvm.lv=rl/root rd.lvm.lv=rl/swap rd.break
initrd ($root)/initramfs-5.14.0-570.55.1.el9_6.x86_64.img $tuned_initrd
```

Рис. 3.12: 11

Чтобы получить доступ к системному образу для чтения и записи пишем mount

```
-o remount,rw /sysroot
```

```
switch_root:/# mount -o remount,rw /sysroot
```

Рис. 3.13: 12

Далее сделаем содержание каталога /sysimage новым корневым каталогом, набрав chroot /sysroot и изменим пароль

```

switch_root:/# chroot /sysroot
sh-5.1# passwd
Изменение пароля пользователя root.
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль должен содержать не менее 8 символов
Повторите ввод нового пароля:
Извините, но пароли не совпадают.
passwd: Ошибка при операциях с маркером проверки подлинности
sh-5.1# passwd
Изменение пароля пользователя root.
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль должен содержать не менее 8 символов
Повторите ввод нового пароля:
passwd: данные аутентификации успешно обновлены.
sh-5.1#

```

Рис. 3.14: 13

Загружаем политику SELinux с помощью команды `load_policy -i`

```

sh-5.1# load_policy -i
[ 187.447875] audit: type=1484 audit(1763111694.886:2): enforcing=1 old_enforcing=0 audit=4294967295 ses=4294967295 enabled=1 old_enabled=1 lsm=selinux
[ 187.473819] SELinux: policy capability network_peer_controls=1
[ 187.473976] SELinux: policy capability open_perms=1
[ 187.474898] SELinux: policy capability extended_socket_class=1
[ 187.474215] SELinux: policy capability always_check_network=0
[ 187.474332] SELinux: policy capability cgroup_scselabel=1
[ 187.474458] SELinux: policy capability nnp_nosuid_transitions=1
[ 187.474566] SELinux: policy capability genfs_scselabel_xpnlinks=1
[ 187.586084] audit: type=1483 audit(1763111694.141:3): audit=4294967295 ses=4294967295 lsm=selinux res=1

```

Рис. 3.15: 14

После чего установи правильный тип контекста для `/etc/shadow` введя команду `chcon -t shadow_t /etc/shadow`

```

sh-5.1# chcon -t shadow_t /etc/shadow
sh-5.1#

```

Рис. 3.16: 15

4 Контрольные вопросы

1. Какой файл конфигурации следует изменить для применения общих изменений в GRUB2?

Ответ - /etc/default/grub

2. Как называется конфигурационный файл GRUB2, в котором вы применяете изменения для GRUB2?

Ответ - /etc/default/grub

3. После внесения изменений в конфигурацию GRUB2, какую команду вы должны выполнить, чтобы изменения сохранились и воспринялись при загрузке системы?

Ответ - update-grub или grub-mkconfig -o /boot/grub/grub.cfg

5 Выводы

После выполнения лабораторной работы я получил навыки работы с загрузчиком системы GRUB2.