



Методическое пособие по выполнению домашнего задания по курсу  
«Администратор Linux. Professional»

## **Домашнее задание Стенд ZFS**

### **1. Введение**

ZFS (Zettabyte File System) — файловая система, созданная компанией Sun Microsystems в 2004-2005 годах для ОС Solaris. Эта файловая система поддерживает большие объёмы данных, объединяет в себе концепции файловой системы, RAID-массивов, менеджера логических дисков и принципы легковесных файловых систем.

ZFS продолжает активно развиваться. К примеру проект FreeNAS использует возможности ZFS для реализации ОС для управления SAN/NAS хранилищ.

Из-за лицензионных ограничений, поддержка ZFS в GNU/Linux ограничена. По умолчанию ZFS отсутствует в ядре linux.

Основное преимущество ZFS — это её полный контроль над физическими носителями, логическими томами и постоянное поддержание консистенции файловой системы. Опираясь на разных уровнях абстракции данных, ZFS способна обеспечить высокую скорость доступа к ним, контроль их целостности, а также минимизацию фрагментации данных. ZFS гибко настраивается, позволяет в процессе работы изменять объём дискового пространства и задавать разный размер блоков данных для разных применений, обеспечивает параллельность выполнения операций чтения-записи

### **2. Цели домашнего задания**

Научится самостоятельно устанавливать ZFS, настраивать пулы, изучить основные возможности ZFS.

### **3. Описание домашнего задания**

1. Определить алгоритм с наилучшим сжатием:
  - Определить какие алгоритмы сжатия поддерживает zfs (gzip, zle, lzjb, lz4);
  - создать 4 файловых системы на каждой применить свой алгоритм сжатия;
  - для сжатия использовать либо текстовый файл, либо группу файлов.
2. Определить настройки пула.  
С помощью команды `zfs import` собрать pool ZFS.  
Командами zfs определить настройки:
  - размер хранилища;
  - тип pool;
  - значение recordsize;
  - какое сжатие используется;
  - какая контрольная сумма используется.
3. Работа со снапшотами:
  - скопировать файл из удаленной директории;

- восстановить файл локально. `zfs receive`;
- найти зашифрованное сообщение в файле `secret_message`.

#### 4. Пошаговая инструкция выполнения домашнего задания

ПК на Unix с 8ГБ ОЗУ или виртуальная машина с включенной Nested Virtualization.

Предварительно установленное и настроенное следующее ПО:

Oracle VirtualBox

([https://www.virtualbox.org/wiki/Linux\\_Downloads](https://www.virtualbox.org/wiki/Linux_Downloads)).

Все дальнейшие действия были проверены при использовании VirtualBox v7.1.6 и системы Ubuntu 24.04. Серьёзные отступления от этой конфигурации могут потребовать адаптации с вашей стороны.

#### Создаем виртуальную машину

Создаём виртуальную машину с Ubuntu 24.04 Server. Помимо системного диска добавляем 8 дополнительных дисков по 512 MB.

Заходим на сервер по SSH. Дальнейшие действия выполняются от пользователя root. Переходим в root пользователя: `sudo -i`

##### 1. Определение алгоритма с наилучшим сжатием

Смотрим список всех дисков, которые есть в виртуальной машине:  
`lsblk`

```
[root@zfs ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
sda	8:0	0	25G	0	disk	
_sda1	8:1	0	1M	0	part	
_sda2	8:2	0	2G	0	part	/boot
_sda3	8:3	0	23G	0	part	
_ubuntu--vg-ubuntu--lv	252:0	0	11.5G	0	lvm	/
sdb	8:16	0	512M	0	disk	
sdc	8:32	0	512M	0	disk	
sdd	8:48	0	512M	0	disk	
sde	8:64	0	512M	0	disk	
sdf	8:80	0	512M	0	disk	
sdg	8:96	0	512M	0	disk	
sdh	8:112	0	512M	0	disk	
sdi	8:128	0	512M	0	disk	

Установим пакет утилит для ZFS:

```
[root@zfs ~]# sudo apt install zfsutils-linux
```

Создаём пул из двух дисков в режиме RAID 1:

```
[root@zfs ~]# zpool create otus1 mirror /dev/sdb /dev/sdc
```

Создадим ещё 3 пула:

```
[root@zfs ~]# zpool create otus2 mirror /dev/sdd /dev/sde
[root@zfs ~]# zpool create otus3 mirror /dev/sdf /dev/sdg
[root@zfs ~]# zpool create otus4 mirror /dev/sdh /dev/sdi
```

Смотрим информацию о пулах: `zpool list`

NAME	SIZE	ALLOC	FREE	CKPOINT	EXPANDSZ	FRAG	CAP	DEDUP
HEALTH	ALTROOT							
otus1	480M	116K	480M	-	-	0%	0%	1.00x
ONLINE	-							
otus2	480M	116K	480M	-	-	0%	0%	1.00x
ONLINE	-							
otus3	480M	146K	480M	-	-	0%	0%	1.00x
ONLINE	-							
otus4	480M	166K	480M	-	-	0%	0%	1.00x
ONLINE	-							

Команда `zpool status` показывает информацию о каждом диске, состоянии сканирования и об ошибках чтения, записи и совпадения хэш-сумм. Команда `zpool list` показывает информацию о размере пула, количеству занятого и свободного места, дедупликации и т.д.

Добавим разные алгоритмы сжатия в каждую файловую систему:

- Алгоритм lzjb: `zfs set compression=lzjb otus1`
- Алгоритм lz4: `zfs set compression=lz4 otus2`
- Алгоритм gzip: `zfs set compression=gzip-9 otus3`
- Алгоритм zle: `zfs set compression=zle otus4`

Проверим, что все файловые системы имеют разные методы сжатия:

```
[root@zfs ~]# zfs get all | grep compression
otus1  compression          lzjb                  local
otus2  compression          lz4                  local
otus3  compression          gzip-9              local
otus4  compression          zle                  local
```

Сжатие файлов будет работать только с файлами, которые были добавлены после включения настройки сжатия.

Скачаем один и тот же текстовый файл во все пулы:

```
[root@zfs ~]# for i in {1..4}; do wget -P /otus$i
https://gutenberg.org/cache/epub/2600/pg2600.converter.log; done
```

...

Проверим, что файл был скачан во все пулы:

```
[root@zfs ~]# ls -l /otus*
/otus1:
total 22096
-rw-r--r-- 1 root root 41123477 Feb  2 08:31 pg2600.converter.log

/otus2:
total 18006
-rw-r--r-- 1 root root 41123477 Feb  2 08:31 pg2600.converter.log
```

```
/otus3:
total 10966
-rw-r--r-- 1 root root 41123477 Feb  2 08:31 pg2600.converter.log
```

```
/otus4:
total 40188
-rw-r--r-- 1 root root 41123477 Feb  2 08:31 pg2600.converter.log
```

Уже на этом этапе видно, что самый оптимальный метод сжатия у нас используется в пуле otus3.  
Проверим, сколько места занимает один и тот же файл в разных пулах и проверим степень сжатия файлов:

```
[root@zfs ~]# zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
otus1     21.7M  330M   21.6M   /otus1
otus2     17.7M  334M   17.6M   /otus2
otus3    10.9M  341M   10.7M   /otus3
otus4     39.4M  313M   39.3M   /otus4
```

```
[root@zfs ~]# zfs get all | grep compressratio | grep -v ref
otus1     compressratio          1.81x          -
otus2     compressratio          2.23x          -
otus3     compressratio          3.65x          -
otus4     compressratio          1.00x          -
```

Таким образом, у нас получается, что алгоритм **gzip-9** самый эффективный по сжатию.

## 2. Определение настроек пула

Скачиваем архив в домашний каталог:

```
[root@zfs ~]# wget -O archive.tar.gz --no-check-certificate
'https://drive.usercontent.google.com/download?id=1MvrcEp-WgAQe57a
DEzxSRalPAwbNN1Bb&export=download'
```

Разархивируем его:

```
[root@zfs ~]# tar -xzf archive.tar.gz
zpoolexport/
zpoolexport/filea
zpoolexport/fileb
[root@zfs ~]#
```

Проверим, возможно ли импортировать данный каталог в пул:

```
[root@zfs ~]# zpool import -d zpoolexport/
pool: otus
id: 6554193320433390805
state: ONLINE
status: Some supported features are not enabled on the pool.
      (Note that they may be intentionally disabled if the
      'compatibility' property is set.)
```

action: The pool can be imported using its name or numeric identifier, though some features will not be available without an explicit 'zpool upgrade'.  
config:

```
otus                ONLINE
  mirror-0          ONLINE
    /root/zpoolexport/filea  ONLINE
    /root/zpoolexport/fileb  ONLINE
```

Данный вывод показывает нам имя пула, тип raid и его состав.

Сделаем импорт данного пула к нам в ОС:

```
[root@zfs ~]# zpool import -d zpoolexport/ otus
[root@zfs ~]# zpool status
pool: otus
state: ONLINE
status: Some supported and requested features are not enabled on
the pool.
  The pool can still be used, but some features are
  unavailable.
action: Enable all features using 'zpool upgrade'. Once this is
done,
  the pool may no longer be accessible by software that does
  not support
  the features. See zpool-features(7) for details.
config:
```

NAME	STATE	READ	WRITE	CKSUM
otus	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
/root/zpoolexport/filea	ONLINE	0	0	0
/root/zpoolexport/fileb	ONLINE	0	0	0

errors: No known data errors

Команда `zpool status` выдаст нам информацию о составе импортированного пула.

Если у Вас уже есть пул с именем `otus`, то можно поменять его имя во время импорта: `zpool import -d zpoolexport/ otus newotus`

Далее нам нужно определить настройки: `zpool get all otus`

Запрос сразу всех параметров файловой системы: `zfs get all otus`

```
[root@zfs ~]# zfs get all otus
NAME  PROPERTY                VALUE
SOURCE
otus  size                    480M
-
otus  capacity                0%
-
```

otus altroot	-
default	
otus health	ONLINE
-	
otus guid	6554193320433390805
-	
otus version	-
default	
otus bootfs	-
default	
otus delegation	on
default	
otus autoreplace	off
default	
otus cachefile	-
default	
otus failmode	wait
default	
otus listsnapshots	off
default	
otus autoexpand	off
default	
otus dedupratio	1.00x
-	
otus free	478M
-	
otus allocated	2.09M
-	
otus readonly	off
-	
otus ashift	0
default	
otus comment	-
default	
otus expandsize	-
-	
otus freeing	0
-	
otus fragmentation	0%
-	
otus leaked	0
-	
otus multihost	off
default	
otus checkpoint	-
-	
otus load_guid	4959715620884454762
-	
otus autotrim	off
default	
otus compatibility	off
default	
otus bcloneused	0
-	
otus bclonesaved	0
-	

otus	bcloneratio	1.00x
-		
otus	feature@async_destroy	enabled
local		
otus	feature@empty_bpobj	active
local		
otus	feature@lz4_compress	active
local		
otus	feature@multi_vdev_crash_dump	enabled
local		
otus	feature@spacemap_histogram	active
local		
otus	feature@enabled_txg	active
local		
otus	feature@hole_birth	active
local		
otus	feature@extensible_dataset	active
local		
otus	feature@embedded_data	active
local		
otus	feature@bookmarks	enabled
local		
otus	feature@filesystem_limits	enabled
local		
otus	feature@large_blocks	enabled
local		
otus	feature@large_dnode	enabled
local		
otus	feature@sha512	enabled
local		
otus	feature@skein	enabled
local		
otus	feature@edonr	enabled
local		
otus	feature@userobj_accounting	active
local		
otus	feature@encryption	enabled
local		
otus	feature@project_quota	active
local		
otus	feature@device_removal	enabled
local		
otus	feature@obsolete_counts	enabled
local		
otus	feature@zpool_checkpoint	enabled
local		
otus	feature@spacemap_v2	active
local		
otus	feature@allocation_classes	enabled
local		
otus	feature@resilver_defer	enabled
local		
otus	feature@bookmark_v2	enabled
local		
otus	feature@redaction_bookmarks	disabled
local		

otus	feature@redacted_datasets	disabled
local		
otus	feature@bookmark_written	disabled
local		
otus	feature@log_spacemap	disabled
local		
otus	feature@livelist	disabled
local		
otus	feature@device_rebuild	disabled
local		
otus	feature@zstd_compress	disabled
local		
otus	feature@draid	disabled
local		
otus	feature@zilsaxattr	disabled
local		
otus	feature@head_errlog	disabled
local		
otus	feature@blake3	disabled
local		
otus	feature@block_cloning	disabled
local		
otus	feature@vdev_zaps_v2	disabled
local		

С помощью команды `grep` можно уточнить конкретный параметр, например:

Размер: `zfs get available otus`

```
[root@zfs ~]# zfs get available otus
NAME  PROPERTY  VALUE  SOURCE
otus  available  350M   -
```

Тип: `zfs get readonly otus`

```
[root@zfs ~]# zfs get readonly otus
NAME  PROPERTY  VALUE  SOURCE
otus  readonly  off     default
```

По типу FS мы можем понять, что позволяет выполнять чтение и запись

Значение recordsize: `zfs get recordsize otus`

```
[root@zfs ~]# zfs get recordsize otus
NAME  PROPERTY  VALUE  SOURCE
otus  recordsize  128K   local
```

Тип сжатия (или параметр отключения): `zfs get compression otus`

```
[root@zfs ~]# zfs get compression otus
NAME  PROPERTY  VALUE  SOURCE
otus  compression  zle     local
```

Тип контрольной суммы: `zfs get checksum otus`

```
[root@zfs ~]# zfs get checksum otus
NAME  PROPERTY  VALUE  SOURCE
otus  checksum  sha256  local
```



### 3. Работа со снимком, поиск сообщения от преподавателя

Скачаем файл, указанный в задании:

```
[root@zfs ~]# wget -O otus_task2.file --no-check-certificate
https://drive.usercontent.google.com/download?id=1wgxjih8YZ-cqLgaZVa0lA3h3Y029c3oI&export=download
```

Восстановим файловую систему из снимка:

```
zfs receive otus/test@today < otus_task2.file
```

Далее, ищем в каталоге /otus/test файл с именем "secret\_message":

```
[root@zfs ~]# find /otus/test -name "secret_message"
/otus/test/task1/file_mess/secret_message
```

Смотрим содержимое найденного файла:

```
[root@zfs ~]# cat /otus/test/task1/file_mess/secret_message
https://otus.ru/lessons/linux-hl/
```

Тут мы видим ссылку на курс OTUS, задание выполнено.

Требуется предоставить список команд по каждому заданию и их выводы.

### 5. Критерий оценивания

Статус «Принято» ставится при выполнении следующих условий:

1. Ссылка на репозиторий GitHub.
2. Bash-скрипт, который будет конфигурировать сервер.
3. Документация по каждому заданию.

Создайте файл README.md и снабдите его следующей информацией:

- название выполняемого задания;
- текст задания;
- описание команд и их вывод;
- особенности проектирования и реализации решения,
- заметки, если считаете, что имеет смысл их зафиксировать в репозитории.

### 6. Рекомендуемые источники

Статья о ZFS

<https://ru.wikipedia.org/wiki/ZFS>

Статья «Что такое ZFS? И почему люди от неё без ума?» -

<https://habr.com/ru/post/424651/>

Официальная документация по Oracle Solaris ZFS

<https://docs.oracle.com/cd/E19253-01/819-5461/gbcya/index.html>

Статья о ZFS (теория и практика)  
<http://xgu.ru/wiki/ZFS>