Lab BSY FILES I

Introduction & Prerequisites

This laboratory is to learn how to:

- Learn how to create, format and mount volumes.
- Learn how to test the performance of volumes and file systems.

The following resources and tools are required for this laboratory session:

- Any modern web browser,
- Any modern SSH client application
- OpenStack Horizon dashboard: https://ned.cloudlab.zhaw.ch
- OpenStack account details: please contact the lab assistant in case you already have not received your access credentials.

Important access credentials:

- Username to login with SSH into VMs in ned.cloudlab.zhaw.ch OpenStack cloud from your laptops
    - **`ubuntu`**

Time

The entire session will take 90 minutes.

Task 0 – Setup

In order to get started, create a VM instance from the standard image with a flavor size of your choice. Access the VM with SSH using the username `ubuntu`. Once you have SSH'ed into the VM, change your shell to a root shell (`su -i`). This will not require you run '`sudo`' in front of every command that requires root/system level privilege[1]. Most of the tools should be installed on the VM, missing ones can be installed using "`apt`".

[1] This is for convenience and not recommended to do on production systems

Task 1 – Basic file and directory operations, link files and analyse i-nodes

In this task you will create some files, copy and move them, change attributes, create hard and soft links (symlink) and give a look to i-nodes.

Start by editing a simple helloworld.c text file in your home directory using an editor like nano. Compile the C helloworld.c program to an executable file named helloworld using gcc and run it.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo Welt\n");
  return 0;
}
```

List the files helloworld.c and helloworld with the ls -al command. Are there any differences concerning file permissions?

Change the permission of helloworld to read and write only. Use the chmod command.

Can you run helloworld now?

Change the permission back. And verify that afterwards helloworld can be executed again.

Create a hard link to the file helloworld.c named hw_hard.c and a soft link with relative path named hw_soft.c also to the file helloworld.c. Use the ln command (man ln).

What happened with the links attribute for the files helloword.c and hw_hard.c?

List the i-node numbers of the files helloword.c, hw_hard.c and hw_soft.c. Use the ls -i command. What do you notice?

Edit the file hw_soft.c with nano and give a look to the file attributes of helloworld.c, hw_hard.c and hw_soft.c. Where are the changes?

Create a subdirectory named mysub in your home directory and move the files hw_hard.c and hw_soft.c in this subdirectory. Use the commands mkdir and mv (see man).

Did the i-node numbers of the files change?

Can you edit hw_hard.c and hw_soft.c in mysub? Why or why not? Use the file command to find an answer (man file).

Create a subdirectory named mymnt in /mnt and mount a new tmpfs filesystem with the command $$sudo mount -t tmpfs -o size=2G tmpfs /mnt/mymnt

Move the file hw_hard.c to the directory /mnt/mymnt. Has the i-node number changed? Why or why not?

What happens with the file helloworld.c in your home directory, if you edit hw_hard.c?

Delete all files in /mnt/mymnt with the rm command, unmount the tmpfs filesystem with the command $$sudo umount /mnt/mymnt and remove the mymnt directory with the rmdir command.

Task 2 – Create, partition and format volumes

In this task you will create two volumes in the ned.cloudlab.zhaw.ch OpenStack environment and attach them to an existing instance. Then you will partition the volumes and format the partitions with different file systems.

Create two volumes of 8 GB each in the ned.cloudlab.zhaw.ch OpenStack environment. To do this, click on the button “Cretae Volume” in the Volume section.

Attach the two volumes to an existing running instance. Select therefore the action “Manage Attachments”.

How can you verify that the volumes are successfully attached?

How can you get partitioning information for the different attached volumes?

What does GPT mean? And how many partitions can be created using this partition scheme?

Create two Linux file system partitions of 2 GB each using gdisk (man gdisk) and verify the result.

Try the commands $ parted -l or $ lsblk to list volumes and partitions. Which of them gives you information about associated mountpoint?

Format the two partitions of the first attached volume with the ext4 file system (man mkfs). The first partition with block size 1024B and the second partition with block site 4096B.

Partition the second volume too and format the partitions with other file systems (ext2, ext3, ...)

Task 3 – Mounting volumes and testing the performance

Create four subdirectories v1, v2, v3 and v4 in your home directory and mount the different partitions to those directories.

Install the customizable tool fio (man fio).

Create the script called fio-benchmark.txt with the following content in your working directory:

```
; -- start job file –
[random-writers]
rw=read
size=50m
numjobs=1
; -- end job file --
```

Interpret these instructions. What is the experiment about?

Run the file with fio ($ fio fio-benchmark.txt) several times form the subdirectories v1, v2 and may be also v3, v4 and iInterpret the results. Is the performance in ext4 with 4096B block size higher? Keep in mind, that we are working in a virtualized environment.

Cleanup!

IMPORTANT: At the end of the lab session:

- **Delete** all OpenStack VMs, volumes, security group rules that are no longer needed. You may want to keep some data for exam preparations.

Additional Documentation

OpenStack Horizon documentation can be found on the following pages:

- User Guide: https://docs.openstack.org/horizon/latest/