

System Administration for Beginners

Week 8 Laboratory

April 5, 2010

1 Introduction

You should work on this laboratory in your project groups. Due to the inherent limitations with the virtualization technology being used, please use the OCF's servers in completing this laboratory. We will be looking more closely at quotas and access control lists (ACLs).

1.1 Submission Instructions

At the top of each submission, please provide the assignment name, your group members names, `inst` logins (`cs198-XX`), and your email addresses. Answer the following questions below. If no specific information is being asked, include any output or answers that you think would help show us that you understand the material (text only). Turn in your *paper* submission at the start of class next week.

2 Quotas

1. Check your quota using `quota`. How much space do you have in your home directory?
2. Fill your home directory past the soft limit, and check your quota again. What's different about the output (aside from how much space you're using)? What does this mean?
3. Attempt to fill your home directory past the hard limit. Is this possible? What happens if you exceed the hard limit?
4. Delete the files so you're no longer over the limits.

3 Access Control Lists

The `man` pages for `getfacl` and `setfacl` will help you greatly with this lab. We recommend you read through it and understand the general principles, and

take a closer look at the examples. Note that the options and parameters to the aforementioned programs will vary depending on the operating system.

As we explored in class, ACLs allow for more fine-grained control over permissions. Because permissions in general only allow for three categories, ACLs allow us to specify more detailed permissions should we require more categories than just the owner, group, and everyone. For example, consider the user `alice` and `bob`. Both users are in the group `ocf` (you can verify this for your login using `groups`), but so is `eve`.

As a regular user, you can't change the group you're in or add new groups. However, `alice` wants to share a file with `bob`, but not `eve`. If the permissions for the group bit is set, it's applicable for all members of the group `ocf`. ACLs allow us to set permissions that the traditional system won't allow for.

NOTE: You should include any relevant commands and output that you used for this lab. For example, if an ACL entry was created, you should write down the command and the resulting output from `getfacl`.

5. Read the `man` pages for `setfacl` and `getfacl`. What needs to be modified on the filesystem so that ACLs are supported?
6. Check the ACL entry for a few files in your home directory. What is the format of the output? Since we did not initially set an ACL entry for this file, where is it getting the values from?
7. Set the file permissions so that only you can read it. Can your partner read the file?
8. Create an ACL entry for your partner so that he or she can read the file. Can your partner read the file, now?
9. What does the `#effective` entry mean in an ACL entry?
10. Modify the ACL entry so that your partner can write to the file. Can your partner write to it?
11. Modify the ACL entry so that you can't write to the file, but your partner can. Can you write to the file? Can your partner write to the file?
12. Modify the ACL entry so that your partner can't read or write to the file, but members of group `ocf` can. Can your partner access the file?
13. Say that for every file you create, you always want to give read and write access to your partner, but no one else. Using `setfacl`, how would you set the default permissions without having to manually run `setfacl` on each file?
14. You want to delete your partner from the ACL entry in a file. How do you do this?
15. Create a new file and, using `|`, set the ACL for this new file using an ACL entry from another file.