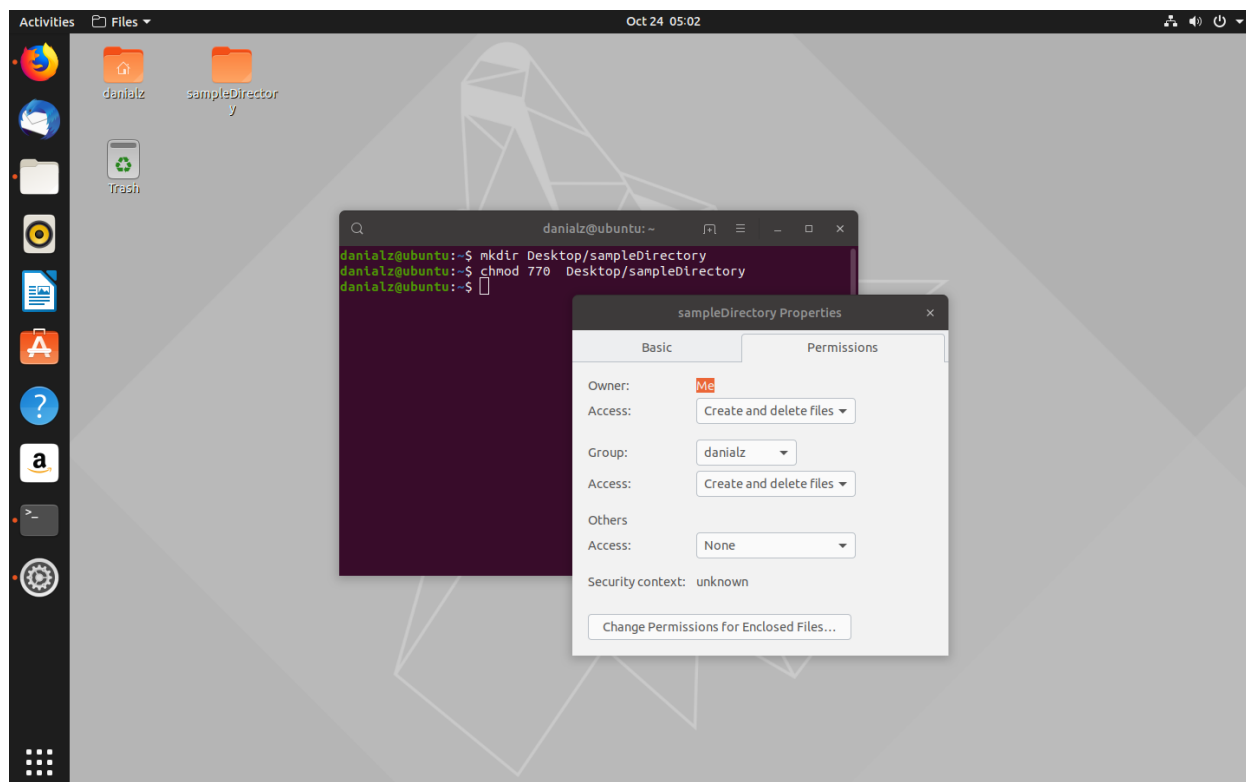
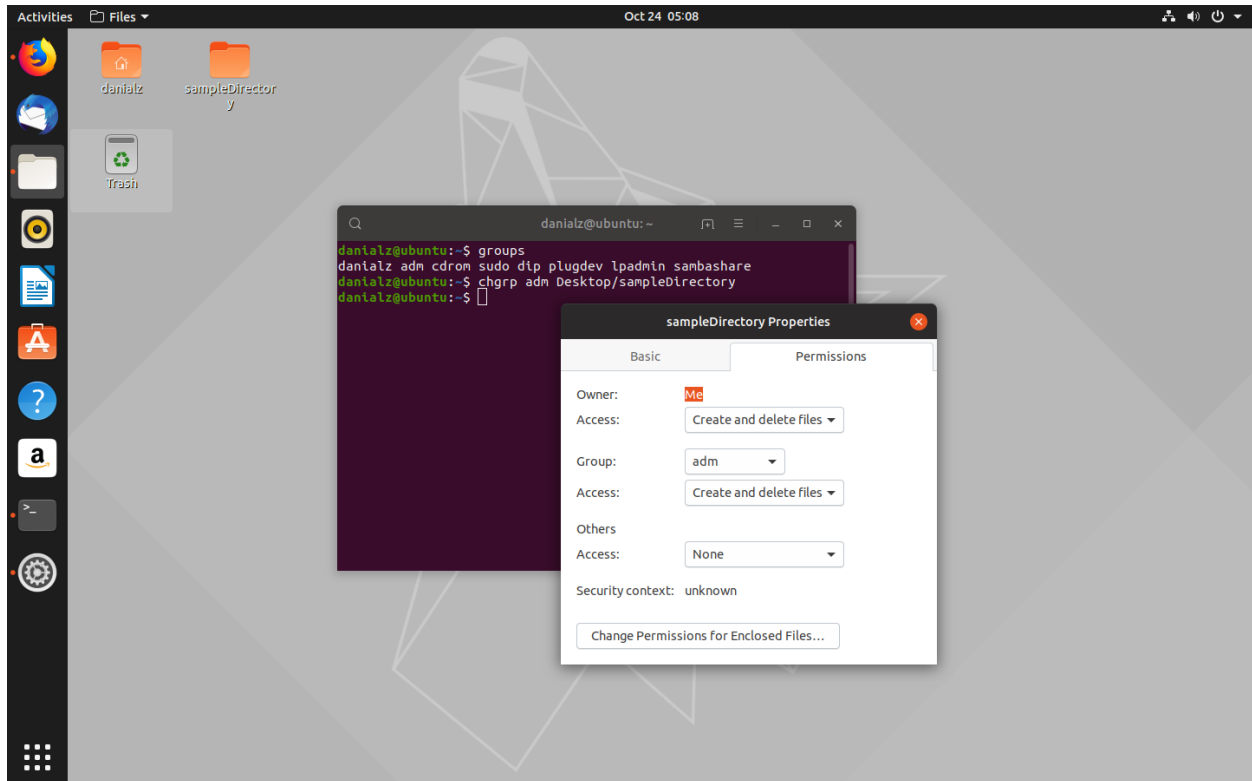


گزارش کار آزمایش اول

۱. از دستور `mkdir` برای ساخت دایرکتوری و از `chmod` برای تغییر دسترسی‌ها استفاده می‌کنیم.



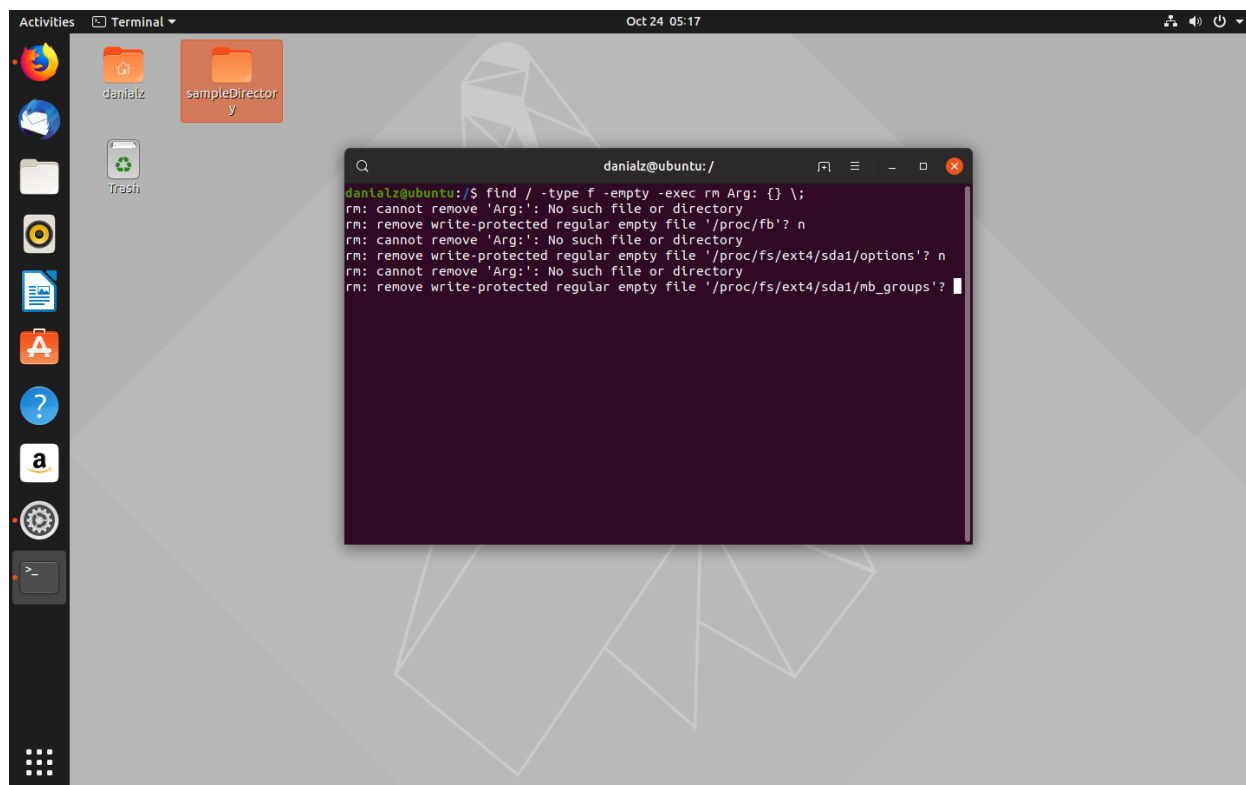
۲. از دو دستور `id -Gn` و `groups` برای گرفتن لیست گروه‌هایی که در آن عضو هستیم، می‌توان استفاده کرد. قابل ذکر است که دستور دوم منسوخ (Deprecated) است. همچنین اطلاعات تمام گروه‌ها و کاربرها در فایل‌های `groups` و `passwd` در آدرس `/etc` قابل مشاهده است. برای تغییر گروه یک فایل هم از دستور `chgrp` استفاده می‌کنیم.



۳. ورودی دستور `chmod` می‌تواند ۴ رقمی باشد؛ در این صورت ۳ رقم آخر مختص دسترسی‌های User, Group, Others هستند. اما رقم اول برای تعیین `suid`, `sgid`, `sticky` bits استفاده می‌شود. عدد ۴ رقم اول به معنی `suid` می‌باشد. `suid` برای تعیین `effective user id` استفاده می‌شود. مثلاً اگر `suid` یک فایل `root` باشد، آن فایل فارغ از آن که چه کسی آن را اجرا کرده، با دسترسی‌های `root` اجرا می‌شود. در ادامه کاربرد `suid` را بررسی می‌کنیم.

گاهی باید به کاربرها اجازه دهیم تا `administrative task` انجام دهند بدون آن که دسترسی مناسب را داشته باشند؛ برای مثال کاربر عادی دسترسی مشاهده‌ی رمز و اطلاعات کاربران را (که در فایل `/etc/passwd` ذخیره شده است) ندارد. اما باید به کاربر اجازه‌ی تغییر رمز خودش را بدهیم. برای این کار به `task` تغییر رمز، `effective user id` مناسب یعنی `root` داده می‌شود.

۴. از دستور find برای پیدا کردن فایل با مشخصات سوال استفاده می‌کنیم. برای پاک کردن این فایل‌ها می‌توانیم از دو سوییچ delete یا exec rm استفاده کنیم. همان‌طور که می‌بینید، بعضی از این فایل‌های خالی، فایل‌های سیستمی هستند و قبل از حذف آن‌ها با هشدار مواجه می‌شویم.



```
danialz@ubuntu:/$ find / -type f -empty -exec rm Arg: {} \;
rm: cannot remove 'Arg:': No such file or directory
rm: remove write-protected regular empty file '/proc/fb'? n
rm: cannot remove 'Arg:': No such file or directory
rm: remove write-protected regular empty file '/proc/fs/ext4/sda1/options'? n
rm: cannot remove 'Arg:': No such file or directory
rm: remove write-protected regular empty file '/proc/fs/ext4/sda1/mnt_groups'?

```