OWNERSHIP DAN PERMISSIONS

7

OBJEKTIF :

1. Mahasiswa Mampu Memahami dan memanipulasi izin file dan pengaturan kepemilikan.
2. Mahasiswa Mampu Memahami Perintah-Perintah yang Digunakan Untuk Memanipulasi Izin File dan Pengaturan Kepemilikan.

PENDAHULUAN

File ownership merupakan hal yang sangat penting untuk keamanan file. Setiap file memiliki pemilik pengguna dan pemilik grup.

Bab ini berfokus pada cara menentukan kepemilikan pengguna dan grup dari sebuah file. Selain itu, konsep izin file dan direktori dieksplorasi, termasuk bagaimana mengubah izin pada file dan direktori. Izin default adalah izin yang diberikan ke file dan direktori saat pertama kali dibuat.

FILE OWNERSHIP

Secara default, pengguna memiliki file yang mereka buat. Meskipun kepemilikan ini dapat diubah, fungsi ini memerlukan hak administratif. Meskipun sebagian besar perintah biasanya menampilkan pemilik pengguna sebagai nama, sistem operasi mengaitkan kepemilikan pengguna dengan UID untuk nama pengguna tersebut.

Setiap file juga memiliki pemilik grup. Secara default, grup utama pengguna yang membuat file adalah pemilik grup dari file baru. Pengguna diizinkan untuk

mengubah pemilik grup dari file yang mereka miliki ke grup mana pun yang mereka ikuti. Mirip dengan kepemilikan pengguna, pengaitan file dengan grup tidak dilakukan secara internal oleh sistem operasi berdasarkan nama, tetapi oleh GID grup.

Karena kepemilikan ditentukan oleh UID dan GID yang terkait dengan file, mengubah UID pengguna (atau menghapus pengguna) memiliki efek membuat file yang awalnya dimiliki oleh pengguna tersebut tidak memiliki pemilik pengguna yang sebenarnya. Jika tidak ada UID di file / etc / passwd yang cocok dengan UID pemilik file, maka UID (nomor) ditampilkan sebagai pemilik pengguna file, bukan nama pengguna (yang sudah tidak ada lagi). Hal yang sama terjadi pada kelompok.

Perintah id dapat berguna untuk memverifikasi akun pengguna mana yang Anda gunakan dan grup mana yang Anda miliki untuk digunakan. Dengan melihat output dari perintah ini, Anda dapat melihat informasi identitas pengguna yang diekspresikan sebagai angka dan nama.

Output dari perintah id menampilkan UID dan nama akun pengguna dari pengguna saat ini diikuti dengan GID dan nama grup dari grup utama dan GID dan nama grup dari semua keanggotaan grup:

```
sysadmin@localhost:~$ id

uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm)
,27(sudo),1005(research),1006(development)
```

Contoh di atas menunjukkan pengguna memiliki UID 1001 untuk sysadmin akun pengguna. Ini juga menunjukkan bahwa grup utama untuk pengguna ini memiliki GID 1001 untuk sysadmin grup.

Karena akun pengguna dan akun grup utama memiliki pengidentifikasi numerik dan nama yang sama, ini menunjukkan bahwa pengguna ini berada di grup pribadi pengguna (UPG). Selain itu, pengguna dalam contoh ini termasuk dalam

empat grup tambahan: grup adm dengan GID 4, grup sudo dengan GID 27, grup riset dengan GID 1005, dan grup pengembangan dengan GID 1006.

Ketika sebuah file dibuat, file itu milik pengguna saat ini dan grup utama mereka saat ini. Jika pengguna dari contoh sebelumnya menjalankan perintah sentuh untuk membuat file, maka pemilik pengguna file tersebut adalah pengguna sysadmin, dan pemilik grup adalah grup sysadmin:

```
sysadmin@localhost:~$ touch /tmp/filetest1
```

File ownership dapat dikonfirmasi menggunakan opsi daftar panjang -l dari perintah ls.

```
sysadmin@localhost:~$ ls -l /tmp/filetest1
-rw-rw-r--. 1 sysadmin sysadmin 0 Oct 21 10:18 /tmp/filetest1
```

File ownership juga berlaku untuk file tersembunyi di sistem. File tersembunyi, yang dimulai dengan titik . karakter terdaftar menggunakan opsi -a dari perintah ls. Dua file tersembunyi pertama yang terdaftar adalah yang terbaru . dan induk .. direktori masing-masing. Kepemilikan semua file dan subdirektori dalam direktori saat ini dapat dicantumkan menggunakan perintah ls -la.

```
sysadmin@localhost:~$ ls -la
total 60
drwxr-xr-x 1 sysadmin sysadmin 4096 Nov  3 22:29 .
drwxr-xr-x 1 root     root     4096 Mar 14  2016 ..
-rw-r--r-- 1 sysadmin sysadmin  220 Apr  3  2012 .bash_logout
-rw-r--r-- 1 sysadmin sysadmin 3768 Mar 14  2016 .bashrc
drwx------ 2 sysadmin sysadmin 4096 Nov  3 22:29 .cache
-rw-r--r-- 1 sysadmin sysadmin  675 Apr  3  2012 .profile
-rw-r--r-- 1 sysadmin sysadmin   74 Mar 14  2016 .selected_editor
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Desktop
```

```
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Documents
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Downloads
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Music
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Pictures
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Public
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Templates
drwxr-xr-x 2 sysadmin sysadmin 4096 Mar 14  2016 Videos
```

Harap Diperhatikan!

Output dari perintah ls -l mencakup banyak informasi yang relevan dengan bab ini termasuk:

- **Permissions**

```
-rw-rw-r--. 1 sysadmin sysadmin 0 Oct 21 10:18 /tmp/filetest1
```

- **User Owner**

```
-rw-rw-r--. 1 sysadmin sysadmin 0 Oct 21 10:18 /tmp/filetest1
```

- **Group Owner**

```
-rw-rw-r--. 1 sysadmin sysadmin 0 Oct 21 10:18 /tmp/filetest1
```

CHANGING GROUP

Jika Anda tahu bahwa file yang akan Anda buat harus dimiliki grup yang berbeda dari grup utama Anda saat ini, maka Anda dapat menggunakan perintah newgrp untuk changing group utama Anda saat ini.

```
newgrp group_name
```

Perintah id mencantumkan informasi identitas Anda, termasuk keanggotaan grup Anda. Jika Anda hanya tertarik untuk mengetahui grup Anda, maka Anda dapat menjalankan perintah grup:

```
sysadmin@localhost:~$ groups

sysadmin adm sudo research development
```

Output dari perintah grup mungkin tidak sedetail output dari perintah id, tetapi jika yang perlu Anda ketahui adalah grup mana yang dapat Anda alihkan dengan menggunakan perintah newgrp, maka perintah grup menyediakan informasi yang Anda butuhkan. Output perintah id memang menunjukkan grup utama Anda saat ini, jadi ini berguna untuk memverifikasi bahwa perintah newgrp berhasil.

Misalnya, jika pengguna sysadmin berencana memiliki file yang dimiliki oleh penelitian grup, tetapi itu bukan grup utama pengguna, maka pengguna dapat menggunakan perintah newgrp dan kemudian memverifikasi grup utama yang benar dengan perintah id sebelum membuat file baru:

```
sysadmin@localhost:~$ id

uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm)
,27(sudo),1005(research),1006(development)
sysadmin@localhost:~$ newgrp research
sysadmin@localhost:~$ id
```

```
uid=1001(sysadmin) gid=1005(research) groups=1005(research),4(adm)
,27(sudo),1001(sysadmin),1006(development)
```

Berdasarkan output dari perintah sebelumnya, awalnya GID pengguna adalah 1001 untuk pengguna sysadmin, kemudian perintah newgrp dijalankan, dan GID utama pengguna menjadi 1005, grup riset. Setelah perintah ini dijalankan, jika pengguna membuat file lain dan melihat detailnya, file baru tersebut akan menjadi milik grup riset:

```
sysadmin@localhost:~$ touch /tmp/filetest2
sysadmin@localhost:~$ ls -l /tmp/filetest2
-rw-r--r--. 1 sysadmin research 0 Oct 21 10:53 /tmp/filetest2
```

Perintah newgrp membuka shell baru; selama pengguna tetap berada di shell itu, grup utama tidak akan berubah. Untuk mengalihkan grup utama kembali ke aslinya, pengguna dapat meninggalkan shell baru dengan menjalankan perintah keluar. Sebagai contoh:

```
sysadmin@localhost:~$ id
uid=1001(sysadmin) gid=1005(research) groups=1005(research),4(adm)
,27(sudo),1001
(sysadmin),1006(development)
sysadmin@localhost:~$ exit
exit
sysadmin@localhost:~$ id
uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm)
,27(sudo),1005(research),1006(development)
```

Harap Diperhatikan!

Hak istimewa administratif diperlukan untuk changing group utama pengguna secara permanen. Pengguna root akan menjalankan perintah berikut:

```
usermod -g groupname username
```

CHANGING GROUP OWNERSHIP

Untuk mengubah pemilik grup dari file yang ada, dapat digunakan perintah chgrp.

```
chgrp group_name file
```

Sebagai pengguna root, perintah chgrp dapat digunakan untuk mengubah pemilik grup dari file apa pun ke grup mana pun. Sebagai pengguna tanpa hak akses administratif, perintah chgrp hanya dapat digunakan untuk mengubah pemilik grup dari file ke grup yang pengguna sudah menjadi anggota:

```
sysadmin@localhost:~$ touch sample
sysadmin@localhost:~$ ls -l sample
-rw-rw-r-- 1 sysadmin sysadmin 0 Oct 23 22:12 sample
sysadmin@localhost:~$ chgrp research sample
sysadmin@localhost:~$ ls -l sample
-rw-rw-r--. 1 sysadmin research 0 Oct 23 22:12 sample
```

Jika pengguna mencoba untuk changing group ownership dari file yang tidak dimiliki pengguna, mereka menerima pesan kesalahan:

```
sysadmin@localhost:~$ chgrp development /etc/passwd
chgrp: changing group of '/etc/passwd': Operation not permitted
```

Untuk changing group ownership dari semua file dari struktur direktori, gunakan opsi -R rekursif ke perintah chgrp. Misalnya, perintah dalam contoh berikut akan changing group ownership dari direktori test_dir dan semua file dan subdirektori dari direktori test_dir.

```
sysadmin@localhost:~$ chgrp -R development test_dir
```

Harap Diperhatikan!

Meskipun Anda dapat melihat file ownership dengan opsi -l ke perintah ls, sistem menyediakan perintah lain yang berguna saat melihat kepemilikan dan izin file: perintah stat. Perintah stat menampilkan informasi yang lebih detail tentang file, termasuk memberikan kepemilikan grup berdasarkan nama grup dan nomor GID:

```
sysadmin@localhost:~$ stat /tmp/filetest1
  File: `/tmp/filetest1'
  Size: 0               Blocks: 0          IO Block: 4096   regular
empty file
Device: fd00h/64768d   Inode: 31477       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1001/sysadmin)   Gid: ( 1001/sys
admin)
Access: 2013-10-21 10:18:02.809118163 -0700
Modify: 2013-10-21 10:18:02.809118163 -0700
Change: 2013-10-21 10:18:02.809118163 -0700
```

CHANGING USER OWNERSHIP

Perintah chown memungkinkan pengguna root untuk changing user ownership atas file dan direktori. Pengguna biasa tidak dapat menggunakan perintah ini untuk mengubah pemilik pengguna dari file, bahkan untuk memberikan kepemilikan salah satu file mereka kepada pengguna lain. Namun, perintah chown juga mengizinkan pengubahan kepemilikan grup, yang dapat dilakukan oleh root atau pemilik file.

Ada tiga cara berbeda untuk menjalankan perintah chown. Metode pertama digunakan untuk mengubah hanya pemilik pengguna file.

```
chown user /path/to/file
```

Misalnya, jika pengguna root ingin changing user ownership atas file abc.txt menjadi pengguna jane, maka perintah berikut dapat dijalankan:

```
root@localhost:~# chown jane /tmp/filetest1
root@localhost:~# ls -l /tmp/filetest1
-rw-rw-r-- 1 jane sysadmin 0 Dec 19 18:44 /tmp/filetest1
```

Metode kedua adalah mengubah pengguna dan grup; ini juga membutuhkan hak akses root. Untuk melakukannya, pisahkan pengguna dan grup dengan tanda titik dua atau karakter titik. Sebagai contoh:

```
chown user:group /path/to/file
chown user.group /path/to/file
```

```
root@localhost:~# chown jane:users /tmp/filetest2
root@localhost:~# ls -l /tmp/filetest2
-rw-r--r-- 1 jane users 0 Dec 19 18:53 /tmp/filetest2
```

Jika pengguna tidak memiliki hak root, mereka dapat menggunakan metode ketiga untuk mengubah pemilik grup dari file seperti perintah chgrp. Untuk menggunakan chown hanya untuk changing group ownership dari file, gunakan titik dua atau titik sebagai awalan untuk nama grup:

```
chown :group /path/to/file
chown .group /path/to/file
```

```
jane@localhost:~$ chown .users /tmp/filetest1
jane@localhost:~$ ls -l /tmp/filetest1
```

```
-rw-rw-r-- 1 jane users 0 Dec 19 18:44 /tmp/filetest1
```

PERMISSIONS

Output dari perintah ls -l menampilkan sepuluh karakter di awal setiap baris. Karakter ini menunjukkan jenis file dan izin file. Misalnya, perhatikan output dari perintah berikut:

```
root@localhost:~# ls -l /etc/passwd

-rw-r--r--. 1 root root 4135 May 27 21:08 /etc/passwd
```

Tipe File

Karakter pertama dari setiap baris menunjukkan jenis file:

```
-rw-r--r-- 1 root root 4135 May 27 21:08 /etc/passwd
```

Tabel berikut menjelaskan nilai yang mungkin untuk jenis file:

Character	Type of the File
-	File biasa, yang mungkin kosong, atau berisi teks atau data biner.
d	File direktori, yang berisi nama file lain dan tautan ke file tersebut.
l	Tautan simbolik adalah nama file yang merujuk (menunjuk) ke file lain.

Character	Type of the File
b	File blok adalah file yang berhubungan dengan blok perangkat keras tempat dimana data dibaca di dalam blok data.
c	File karakter adalah file yang berhubungan dengan perangkat keras karakter di mana data dibaca satu byte dalam satu waktu.
p	File pipa berfungsi mirip dengan simbol pipa, memungkinkan output dari satu proses untuk berkomunikasi dengan proses lain melalui file pipa, di mana output dari satu proses digunakan sebagai input untuk proses lainnya.
s	File soket memungkinkan dua proses untuk berkomunikasi, di mana kedua proses diizinkan untuk mengirim atau menerima data.

Harap Diperhatikan!

Meskipun semua jenis file tercantum dalam tabel di atas, biasanya Anda tidak menemukan apa pun kecuali file direktori dan tautan biasa kecuali Anda menjelajahi direktori / dev.

Permission Groups

Sembilan karakter berikutnya menunjukkan izin file.

```
-rw-r--r-- 1 root root 4135 May 27 21:08 /etc/passwd
```

Izin yang ditetapkan pada file ini menentukan tingkat akses yang dimiliki pengguna pada file tersebut. Ketika pengguna menjalankan program dan program mengakses file, maka izin diperiksa untuk menentukan apakah pengguna memiliki hak akses yang benar ke file.

Izin dikelompokkan menjadi tiga peran berbeda, mewakili pengguna berbeda yang mungkin mencoba mengakses file.

Jika Anda bukan pemilik dan bukan anggota grup file / direktori, maka izin Anda adalah orang lain(other).

User Owner

```
-rw-r--r-- 1 root root 4135 May 27 21:08 /etc/passwd
```

Karakter 2-4 menunjukkan izin untuk pengguna yang memiliki file tersebut. Jika Anda adalah pemilik file, maka hanya izin pemilik pengguna yang digunakan untuk menentukan akses ke file itu.

Group Owner

```
-rw-r--r-- 1 root root 4135 May 27 21:08 /etc/passwd
```

Karakter 5-7 menunjukkan izin untuk grup yang memiliki file tersebut. Jika Anda bukan pemilik tetapi adalah anggota grup yang memiliki file tersebut, maka hanya izin pemilik grup yang digunakan untuk menentukan akses ke file itu.

Other Permissions

```
-rw-r--r--  1 root root 4135 May 27 21:08 /etc/passwd
```

Karakter 8-10 menunjukkan izin untuk orang lain atau yang kadang-kadang disebut sebagai izin global. Grup ini mencakup semua pengguna yang bukan pemilik file atau anggota grup file.

Tipe Perizinan

Setiap grup diberi tiga tipe dasar perizinan: baca, tulis, dan eksekusi.

User Owner			Group Owner			Other		
Read	Write	Execute	Read	Write	Execute	Read	Write	Execute
r	w	x	r	w	x	r	w	x

Izin itu sendiri tampak sederhana dan memiliki arti yang berbeda tergantung pada apakah mereka diterapkan ke file atau direktori.

Read

Karakter pertama dari setiap grup mewakili izin baca. Terdapat karakter r jika grup tersebut memiliki izin baca, atau karakter - jika grup tidak.

1. Pada file, ini memungkinkan proses untuk membaca konten file, yang berarti konten dapat dilihat dan disalin.
2. Di direktori, nama file dalam direktori dapat dicantumkan, tetapi detail lainnya tidak tersedia.

Write

Karakter kedua dari setiap grup mewakili izin menulis. Ada karakter w jika grup memiliki izin menulis, atau karakter - jika grup tidak.

1. Sebuah file dapat ditulis oleh proses, sehingga perubahan pada file dapat disimpan. Perhatikan bahwa izin w benar-benar membutuhkan izin r pada file untuk bekerja dengan benar.
2. Di direktori, file dapat ditambahkan atau dihapus dari direktori. Perhatikan bahwa izin w membutuhkan izin x pada direktori untuk bekerja dengan benar.

Execute

Karakter ketiga dari setiap grup mewakili izin eksekusi. Ada karakter x jika grup memiliki izin eksekusi, atau karakter - jika grup tidak.

1. Sebuah file dapat dieksekusi atau dijalankan sebagai suatu proses.
2. Pada direktori, pengguna dapat menggunakan perintah cd untuk "masuk ke" direktori dan menggunakan direktori dalam nama jalur untuk mengakses file dan, kemungkinan besar, subdirektori di bawah direktori ini.

UNDERSTANDING PERMISSIONS

Deskripsi jenis izin dapat berguna, tapi tidak efektif, mereka tidak memberikan deskripsi yang jelas tentang cara kerja izin. Untuk lebih memahami cara kerja izin, pertimbangkan skenario berikut ini.

Untuk memahami skenario ini, Anda harus memahami diagram berikut ini:

```
drwxr-xr-x. 17 root root 4096 23:38 /
drwxr-xr-x. 10 root root 128  03:38 /data
-rwxr-xr--.  1 bob  bob  100  21:08 /data/abc.txt
```

Informasi yang relevan disorot. Baris pertama mewakili direktori /, dengan pemilik pengguna root, pemilik grup root, dan izin rwxr-xr-x. Baris kedua

mewakili direktori / data, direktori yang berada di bawah direktori /. Baris ketiga mewakili file abc.txt, yang disimpan di direktori / data.

Skenario #1 – Akses Direktori

Pertanyaan: Berdasarkan informasi berikut, akses apa yang akan dimiliki bob pengguna pada file abc.txt?

```
drwxr-xr-x. 17 root root 4096 23:38 /
drwxr-xr--. 10 root root 128  03:38 /data
-rwxr-xr--.  1 bob  bob  100  21:08 /data/abc.txt
```

Jawaban: Tidak ada.

Penjelasan: Awalnya terlihat bahwa bob pengguna dapat melihat isi dari file abc.txt serta menyalin file tersebut, memodifikasi isinya dan menjalankannya seperti program. Kesimpulan yang salah ini akan menjadi hasil dari melihat hanya pada izin file (rwx untuk pengguna bob dalam kasus ini).

Namun, untuk melakukan apapun dengan file tersebut, pengguna harus terlebih dahulu "masuk" ke direktori / data. Izin untuk bob untuk direktori / data adalah izin untuk "orang lain" (r--), yang berarti bob bahkan tidak bisa menggunakan perintah cd untuk masuk ke direktori. Jika izin eksekusi (--x) ditetapkan untuk direktori, maka pengguna bob akan dapat "masuk" ke direktori, yang berarti izin file itu sendiri akan berlaku.

Hal yang Dipelajari: Izin semua direktori induk harus dipertimbangkan sebelum mempertimbangkan izin pada file tertentu.

Skenario #2 – Melihat Isi Direktori

Pertanyaan: Berdasarkan informasi berikut, siapa yang dapat menggunakan perintah ls untuk menampilkan konten direktori / data (ls / data)?

```
drwxr-xr-x. 17 root root 4096 23:38 /
```

```
drwxr-xr--. 10 root root 128  03:38 /data
-rwxr-xr--.  1 bob  bob  100  21:08 /data/abc.txt
```

Jawaban: Semua pengguna.

Penjelasan: Semua yang dibutuhkan untuk dapat melihat isi direktori adalah ijin pada direktori (dan kemampuan untuk mengakses direktori induk). Izin x untuk semua pengguna di direktori / berarti semua pengguna dapat menggunakan / sebagai bagian dari sebuah jalur, sehingga semua orang dapat melalui direktori / untuk masuk ke direktori / data. Izin r untuk semua pengguna di direktori / data berarti semua pengguna dapat menggunakan perintah ls untuk melihat isinya. Ini termasuk file tersembunyi, jadi perintah ls -a juga berfungsi di direktori ini. Namun, perhatikan bahwa untuk melihat detail file (ls -l), direktori juga memerlukan izin x. Jadi sementara pengguna root dan anggota grup root memiliki akses ini di direktori / data, tidak ada pengguna lain yang dapat mengeksekusi ls -l / data.

Hal yang dipelajari: Izin r memungkinkan pengguna untuk melihat daftar direktori.

Skenario #3 – Menghapus Isi Direktori

Pertanyaan: Berdasarkan informasi berikut, siapa yang dapat menghapus file /data/abc.txt?

```
drwxr-xr-x. 17 root root 4096 23:38 /
drwxrw-rw-. 10 root root 128  03:38 /data
-rwxr-xr--.  1 bob  bob  100  21:08 /data/abc.txt
```

Jawaban: Hanya pengguna root.

Penjelasan: Pengguna tidak membutuhkan izin sama sekali pada file itu sendiri untuk menghapus file. Izin w pada direktori tempat file disimpan diperlukan untuk menghapus file dalam direktori. Berdasarkan itu, tampaknya semua

pengguna dapat menghapus file /data/abc.txt, karena setiap orang memiliki izin w pada direktori.

Namun, untuk menghapus file, Anda juga harus bisa "masuk" ke direktori. Karena hanya pengguna root yang memiliki izin x pada direktori / data, hanya root yang dapat "masuk" ke direktori tersebut untuk menghapus file di direktori ini.

Hal yang dipelajari: Izin w memungkinkan pengguna untuk menghapus file dari direktori, tetapi hanya jika pengguna juga memiliki izin x pada direktori.

Skenario #4 – Mengakses Isi Direktori

Pertanyaan: Benar atau Salah: Berdasarkan informasi berikut, bob pengguna dapat berhasil menjalankan perintah berikut: more /data/abc.txt?

```
drwxr-xr-x. 17 root root 4096 23:38 /

dr-xr-x--x. 10 root root 128  03:38 /data

-rwxr-xr--.  1 bob  bob  100  21:08 /data/abc.txt
```

Jawaban: Benar.

Penjelasan: Seperti yang disebutkan sebelumnya, untuk mengakses file, pengguna harus memiliki akses ke direktori. Akses ke direktori hanya membutuhkan izin x; meskipun izin r akan berguna untuk membuat daftar file dalam direktori, itu tidak diperlukan untuk "masuk ke" direktori dan mengakses file di dalam direktori.

Ketika perintah more /data/abc.txt dijalankan, izin berikut diperiksa: x izin pada direktori /, x izin pada direktori data dan r izin pada file abc.txt. Karena bob pengguna memiliki semua izin ini, perintah berhasil dijalankan.

Hal yang dipelajari: Izin x diperlukan untuk "masuk ke" direktori, tetapi izin r pada direktori tidak diperlukan kecuali Anda ingin mencantumkan isi direktori.

Skenario #5 - Kompleksitas Pengguna dan Grup

Pertanyaan: Benar atau Salah: Berdasarkan informasi berikut, bob pengguna dapat berhasil menjalankan perintah berikut: more /data/abc.txt?

Perhatikan bahwa / data direktori memiliki pengguna dan pemilik grup yang berbeda dari contoh sebelumnya.

```
drwxr-xr-x. 17 root root    4096 23:38 /
dr-xr-x---. 10 sue  payroll 128  03:38 /data
-rwxr-xr--.  1 bob  bob     100  21:08 /data/abc.txt
```

Jawaban: Tidak cukup informasi untuk menentukan.

Penjelasan: Untuk mengakses file /data/abc.txt, bob pengguna harus bisa "masuk ke" direktori / data. Ini memerlukan izin x, yang mungkin dimiliki atau tidak dimiliki bob, bergantung pada apakah dia anggota grup penggajian.

Jika bob adalah anggota dari grup penggajian, maka hak aksesnya pada direktori / data adalah r-x, dan perintah more akan berhasil dijalankan (bob juga membutuhkan x on / dan r pada abc.txt, yang sudah dia miliki).

Jika dia bukan anggota grup penggajian, izinnya pada direktori / data adalah ---, dan perintah lainnya akan gagal.

Hal yang Dipelajari: Anda harus melihat setiap izin file dan direktori secara terpisah dan mengetahui di grup mana akun pengguna tersebut berada.

Skenario #6 - Prioritas Izin

Pertanyaan: Benar atau Salah: Berdasarkan informasi berikut, bob pengguna dapat berhasil menjalankan perintah berikut: more /data/abc.txt?

Perhatikan bahwa / data direktori memiliki pengguna dan pemilik grup yang berbeda dari contoh sebelumnya

```
drwxr-xr-x. 17 root root 4096 23:38 /

dr-xr-x---. 10 bob  bob  128  03:38 /data

----rw-rwx.  1 bob  bob  100  21:08 /data/abc.txt
```

Jawaban: Salah.

Penjelasan: Ingatlah bahwa jika Anda adalah pemilik file, maka satu-satunya izin yang diperiksa adalah izin pemilik pengguna. Dalam kasus ini, itu akan menjadi --- untuk bob pada file /data/abc.txt.

Dalam kasus ini, anggota grup bob dan "lainnya" memiliki lebih banyak izin pada file daripada yang dimiliki bob.

Hal yang Diperoleh: Jangan memberikan izin kepada pemilik grup dan "orang lain" tanpa menerapkan setidaknya tingkat akses yang sama ke pemilik file.

CHANGING PERMISSIONS

Perintah chmod (mode perubahan) digunakan untuk mengubah izin pada file dan direktori. Ada dua teknik yang dapat digunakan dengan perintah ini: simbolik dan numerik. Kedua teknik tersebut menggunakan sintaks dasar berikut:

```
chmod new_permission file_name
```

Penting: Untuk mengubah izin file, Anda perlu memiliki file atau masuk sebagai pengguna root.

Perhatikan Contoh Berikut:

```
root@localhost:~# touch abc.txt

root@localhost:~# ls -l abc.txt

-rw-r--r-- 1 root root 0 Dec 19 18:58 abc.txt
```

Metode Simbolik

Jika Anda ingin mengubah beberapa izin saat ini, metode simbolik biasanya lebih mudah digunakan. Dengan metode ini, Anda menentukan izin mana yang ingin Anda ubah pada file, dan izin lainnya tetap apa adanya.

Saat menentukan argumen new_permission dari perintah chmod menggunakan metode simbolik, tiga jenis informasi diperlukan.

Mulailah dengan menggunakan satu atau beberapa karakter berikut ini untuk menunjukkan grup izin mana yang akan menerapkan perubahan ke:

u	user owner (Pemilik Pengguna)
g	group owner (Pemilik Grup)
o	others
a	all (user owner, group owner, and others)

Lalu pilih salah satu operator berikut untuk menunjukkan cara mengubah izin:

+	add
-	remove
=	equals

Terakhir, gunakan karakter berikut untuk menentukan jenis izin yang akan diubah:

r	read
w	write
x	execute

Misalnya, untuk memberikan izin tulis kepada pemilik grup pada file bernama abc.txt, Anda dapat menggunakan perintah berikut:

```
root@localhost:~# chmod g+w abc.txt
root@localhost:~# ls -l abc.txt
-rw-rw-r-- 1 root root 0 Dec 19 18:58 abc.txt
```

Hanya izin pemilik grup yang diubah. Semua izin lainnya tetap seperti sebelum eksekusi perintah chmod.

Anda dapat menggabungkan nilai untuk membuat beberapa perubahan pada izin file. Misalnya, pertimbangkan perintah berikut yang menambahkan izin eksekusi ke pemilik pengguna dan pemilik grup dan menghapus izin baca untuk orang lain:

```
root@localhost:~# chmod ug+x,o-r abc.txt
root@localhost:~# ls -l abc.txt
-rwxrwx--- 1 root root 0 Dec 19 18:58 abc.txt
```

Terakhir, Anda dapat menggunakan karakter =, yang menambahkan izin tertentu dan menyebabkan dihapusnya izin yang tidak disebutkan. Misalnya, untuk memberikan izin baca dan eksekusi hanya kepada pemilik pengguna, hapus izin tulis:

```
root@localhost:~# chmod u=rx abc.txt
root@localhost:~# ls -l abc.txt
```

```
-r-xrwx--- 1 root root 0 Dec 19 18:58 abc.txt
```

Metode Numerik

Metode numerik (juga disebut metode oktal) berguna saat mengubah banyak izin pada file. Ini didasarkan pada sistem penomoran oktal di mana setiap jenis izin diberi nilai numerik:

4	Read
2	Write
1	Execute

Dengan menggunakan kombinasi angka dari 0 hingga 7, kombinasi apa pun yang mungkin dari izin baca, tulis, dan eksekusi dapat ditentukan untuk satu kumpulan grup izin. Sebagai contoh:

7	rwx
6	rw-
5	r-x
4	r--
3	-wx

2	-w-
1	--x
0	---

Argumen new_permission ditetapkan sebagai tiga angka, satu angka untuk setiap grup izin. Ketika metode numerik digunakan untuk mengubah izin, kesembilan izin harus ditentukan. Karenanya, metode simbolik umumnya lebih mudah untuk mengubah beberapa izin sedangkan metode numerik lebih baik untuk perubahan yang lebih drastis.

Misalnya, untuk mengatur izin file bernama abc.txt menjadi rwxr-xr-- Anda dapat menggunakan perintah berikut:

```
root@localhost:~# chmod 754 abc.txt
root@localhost:~# ls -l abc.txt
-rwxr-xr-- 1 root root 0 Dec 19 18:58 abc.txt
```

Default Permissions

Perintah *umask* adalah fitur yang digunakan untuk menentukan izin default yang ditetapkan saat file atau direktori dibuat. Izin default ditentukan ketika nilai umask dikurangi dari izin default maksimum yang diizinkan. Izin default maksimum berbeda untuk file dan direktori:

File	rw-rw-rw-
Direktori	rwxrwxrwx

Izin yang awalnya ditetapkan pada file saat dibuat tidak dapat melebihi rw-rw-rw-. Untuk mengatur izin eksekusi pada file, Anda harus membuat file terlebih dahulu, lalu mengubah izinnya.

Perintah umask dapat digunakan untuk menampilkan nilai *umask* saat ini:

```
sysadmin@localhost:~$ umask

0002
```

Rincian output:

• Angka 0 pertama menunjukkan bahwa *umask* diberikan sebagai bilangan oktal.

• Angka 0 kedua menunjukkan izin mana yang akan dikurangi dari izin pemilik pengguna default.

• 0 ketiga menunjukkan izin mana yang akan dikurangi dari izin pemilik grup default.

• Angka terakhir 2 menunjukkan izin mana yang akan dikurangi dari izin default lainnya.

Perhatikan bahwa pengguna yang berbeda mungkin memiliki payung yang berbeda. Biasanya pengguna root memiliki *umask* yang lebih ketat daripada akun pengguna biasa:

```
root@localhost:~# umask

0022
```

Untuk memahami cara kerja *umask*, asumsikan bahwa *umask* sebuah file disetel ke 027 dan perhatikan hal berikut:

File Default	666

Umask	-027
Hasil	640

Umask 027 berarti bahwa file baru akan menerima 640 atau rw-r ----- izin secara default, seperti yang ditunjukkan di bawah ini:

```
sysadmin@localhost:~$ umask 027
sysadmin@localhost:~$ touch sample
sysadmin@localhost:~$ ls -l sample
-rw-r-----. 1 sysadmin sysadmin 0 Oct 28 20:14 sample
```

Karena izin default untuk direktori berbeda dengan file, umask 027 akan menghasilkan izin awal yang berbeda pada direktori baru:

Direktori Default	777
Umask	-027
Hasil	750

Umask 027 berarti bahwa file direktori akan menerima 750 atau rwxr-x --- izin secara default, seperti yang ditunjukkan di bawah ini:

```
sysadmin@localhost:~$ umask 027
sysadmin@localhost:~$ mkdir test-dir
sysadmin@localhost:~$ ls -ld test-dir
drwxr-x---. 1 sysadmin sysadmin 4096 Oct 28 20:25 test-dir
```

Umask baru hanya diterapkan ke file dan direktori yang dibuat selama sesi itu. Saat shell baru dimulai, umask default akan kembali berlaku.

Mengubah *umask* pengguna secara permanen membutuhkan modifikasi file .bashrc yang terletak di direktori home pengguna tersebut.

Referensi:

https://www.netacad.com/courses/os-it/ndg-linux-essentials