

Permission, Link, Pipe, Redirection

Praktikum Sistem Operasi

Ilmu Komputer IPB

2017

Ownership

- ▶ Tiap *file* memiliki *owner*
 - ▶ hanya *superuser* yang dapat mengubah kepemilikan *file*
- ▶ Tiap *file* memiliki *permission*
 - ▶ mengatur hak akses *file* tersebut

Permission

- ▶ Tiga jenis *permission*:

<i>Permission</i>	<i>File</i>	<i>Directory</i>
r	<i>read</i>	<i>list files</i>
w	<i>write</i>	<i>add or remove files</i>
x	<i>execute</i>	<i>enter the directory</i>

- ▶ Tiga jenis *user*:
 - ▶ *user owner* (u)
 - ▶ *group owner* (g)
 - ▶ *others* (o)

unix permissions

JULIA EVANS
@b0rk

<p>3 kinds of things you can do to a file</p> <p>↓ read ↓ write ↓ execute</p>	<pre>\$ ls -l awesome.png</pre> <p>rw- rw- r-- bork staff</p> <p>↑ ↑ ↖</p> <p>bork can do this (user) staff can do this (group) ANYONE can do this</p>
<pre>\$ ls -l /bin/ping</pre> <p>rw^sr-xr-x root root</p> <p>↑</p> <p>setuid flag</p> <p>This means ping <u>always</u> runs as root (who owns it), no matter who started ping</p>	<div data-bbox="639 448 880 789"><p>what's this 755 business?</p><p>7 means rwx 6 → rw- 5 → r-x 4 → r--</p><p>it's binary ? 5 → 101 → r-x</p><p>755 means rwx r-x r-x</p></div> <div data-bbox="886 448 1116 789"><p>more weird permissions things</p><p>setgid</p><p>sticky bit</p><p>but I ran out of space</p></div>

Gambar 1: UNIX permissions

su

Berubah menjadi *user* lain atau menjadi *superuser*.

`su [OPTION] [USERNAME]`

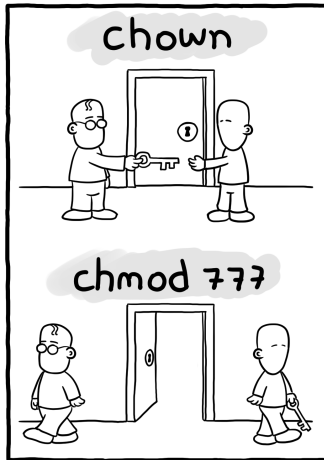
- ▶ `-c COMMAND`: menjalankan perintah `COMMAND`
- ▶ `-l`: seperti *login* langsung (*default environment*)

chown

Mengubah kepemilikan suatu *file*.

```
chown [OPTION] [OWNER] [:GROUP] FILE
```

- ▶ -R: rekursif



Daniel Stori {turnoff.us}

Gambar 2: Membuka akses untuk publik

chmod

Mengganti mode *permission* suatu *file*.

```
chmod [OPTION] MODE[,MODE]... FILE...
```

```
chmod [OPTION] OCTAL-MODE FILE...
```

- ▶ -R: rekursif

Format chmod

- ▶ Format mode simbolis:
 - ▶ [ugoa] [+ -=] [rwxX]
- ▶ Format mode numerik:
 - ▶ digit oktal = 4 (*read*) + 2 (*write*) + 1 (*exec*)
- ▶ *Catatan*: opsi *permission* X hanya akan mengeset bit *execute* untuk direktori saja

Contoh chmod

- ▶ `r--r--r--`

- ▶ `chmod a=r FILE`
- ▶ `chmod 444 FILE`

- ▶ `rw-rw----`

- ▶ `chmod ug=rw,o= FILE`
- ▶ `chmod 660 FILE`

- ▶ `rwxr-xr-x`

- ▶ `chmod a=rx,u+w FILE`
- ▶ `chmod 755 FILE`

Link

1. *Hard link*

- ▶ mengacu pada nomor indeks *file* (inode)
- ▶ tidak terpengaruh terhadap perubahan nama *file*
- ▶ namun hanya bisa dalam satu partisi yang sama

2. *Symbolic link*

- ▶ mengacu pada nama *file*
- ▶ bisa lintas partisi
- ▶ bisa membuat *link* ke direktori
- ▶ namun jika nama *file* yang dirujuk berubah akan mengakibatkan *broken link*

What's an inode and why should I care?

julia evans
@b0rk

drawings.jvns.ca



Gambar 3: Inode

directories + symlinks

@b0rk
Julia Evans

drawings.jvns.ca

What's a directory?

<u>filename</u>	<u>inode number</u>
awesome.jpg	279932
blah.txt	13227
cumberbatch	238333



I made a directory
with 2,000,000
files

It's so
SLOW

listing
your directory
is gonna be
REAL SLOW

(a few seconds at least)

what's a symlink?

it's just a file with the
name of another file in it!

\$ **readlink** my-cool-link
/home/julia/long-complicated-
file-name



OLD
on ext 2 even opening
files in big directories
is slow :(

that's right! ext 2
directories have no index
so you have to **SEARCH**
THE WHOLE THING :(



ext 2 is OLD though. ext3 is
OK.

Gambar 4: Direktori dan symlink

ln

Membuat *link* antar-*file*.

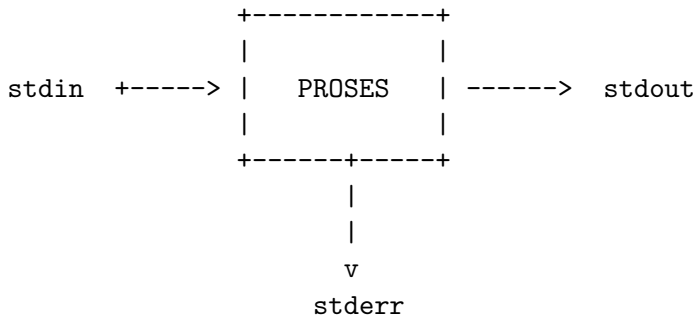
```
ln [OPTION] TARGET LINK-NAME
```

- ▶ `-s`: *symbolic link*

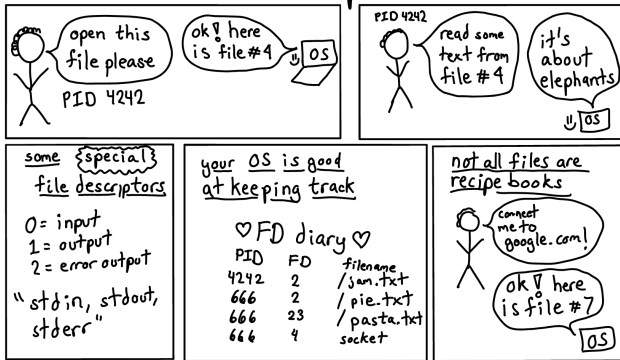
Stream Standar

Setiap proses yang berjalan memiliki tiga *stream* standar I/O:

- ▶ *standard input* (stdin)
- ▶ *standard output* (stdout)
- ▶ *standard error* (stderr)



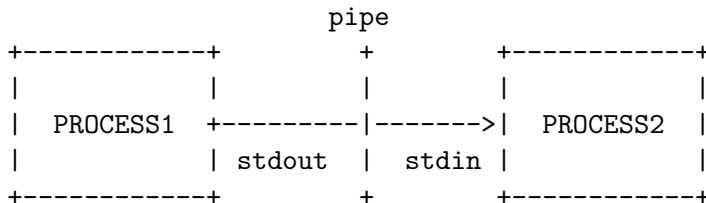
let's learn about ♥ file descriptors ♥



Gambar 5: File descriptor

Pipe

- ▶ Menyalurkan *output* proses menjadi *input* proses selanjutnya
- ▶ Berguna untuk membuat *pipeline* perintah

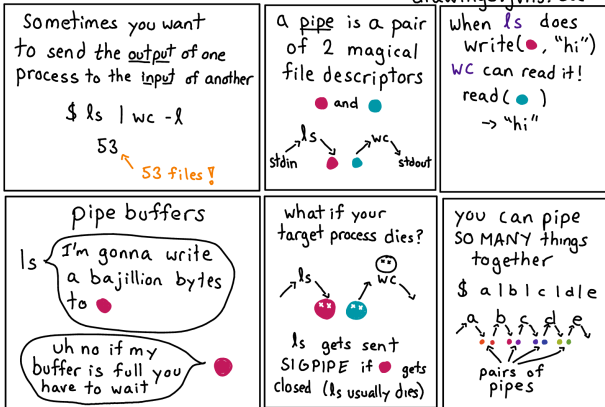


- ▶ Contoh:
 - ▶ `echo "halo" | rev`
 - ▶ `echo "2 + 5" | bc`
 - ▶ `who | wc -l`

pipes

JULIA EVANS
@b0rk

drawings.jvns.ca



Gambar 6: Pipe

Redirect

- Mengarahkan *stream* standar proses ke suatu *file* yang ditentukan oleh pengguna

Karakter	<i>Redirect</i>
<	stdin
>	stdout
>>	stdout (<i>append</i>)
2>	stderr

- Contoh:

```
date > now.txt 2> err.txt  
rev < now.txt  
rev < now.txt > rev.txt
```