

Access permissions and command find.

Jan Trdlička

Department of Computer Systems
Faculty of Information Technology
Czech Technical University in Prague

trdlicka@fit.cvut.cz

December 3, 2018

Contents

- 1 Discussion of homework
- 2 User account
- 3 Access permissions
- 4 Default access permissions
- 5 Command `find`
- 6 Homework

Discussion of homework

- Use the output from the command `ps -eo user,rss,comm`.
How to print the following information for every user that is working on the server?

`user_name`, `nproc=number_of_running_processes`,
`rss=size_of_RSS_memory_alocatted_by_these_processes`

- Example of output

```
$> ps -eo user,rss,comm | nawk -f ps4.awk
```

User	NProc	RSS (KB)

kolensta	2	28888
root	94	689168
trdlicka	4	24448
klamovik	4	38112
...		

Discussion of homework

```
BEGIN {
    printf("User\t\tNProc\tRSS (KB)\n");
    printf("-----\n");
}
{
    if ( NR > 1 ) {
        nproc[$1]=nproc[$1]+1;
        rss[$1]=rss[$1]+$2;
    };
}
END {
    for (i in nproc) {
        printf("%s\t\t%d\t%s\n", i, nproc[i], rss[i]);
    };
}
```

- What information must the OS remember about every user?
 - user name
 - password
 - UID
 - primary GID and secondary GID's
 - home directory
 - login shell
- Where the previous information are stored?
 - in local files: `/etc/passwd` , `/etc/shadow`, `/etc/group`
 - on remote server (NIS, NIS+, LDAP)

Access permissions

- How to determine effective and real user identity of your shell?

```
ps -eo pid,user,ruser,comm | grep "^ *$$"
```

```
ps -eo pid,user,ruser,comm | grep "^ *$$" # in Solaris
```

- How to determine effective and real group identity of your shell?

```
ps -eo pid,group,rgroup,comm | grep "^ *$$"
```

```
ps -eo pid,group,rgroup,comm | grep "^ *$$" # in Solaris
```

- How to determine the owner and owner group of the file /usr/bin/passwd?

```
ls -l /usr/bin/passwd
```

- How to print only access permissions of the directory /etc? What is the meaning of these permissions?

```
ls -ld /etc | cut -c2-10
```

- User has permissions: `rwX`.
- Group and other have permissions: `r-X`.

Access permissions

- Login to server `fray1.fit.cvut.cz`.
- What permissions has the directory `/bin` and what does it mean?

```
ls -ld /bin
```

- `/bin` is symbolik link → use one of the following commands

```
ls -ld /usr/bin
```

```
ls -lLd /bin
```

- Owner `root` has all permissions (read/write/execute).
- Owner group `bin` and other have not write permission. They cannot create/delete subdirectories/files below the directory `/bin (/usr/bin)`.

Access permissions

- Create the following directory structure in your home directory and setup the same permissions (files passwd and date are copies of /etc/passwd and /bin/date, respectively). You and your primary group will be the owner and owner group.

```
dr-x----- user group ps1-09/  
dr-x----- user group ps1-09/A  
-r-x----- user group ps1-09/A/date  
-r-x----- user group ps1-09/A/passwd
```

```
cd ~  
mkdir -p ps1-09/A  
  
cp /etc/passwd ps1-09/A  
cp /bin/date ps1-09/A  
  
chmod -R 500 ps1-09
```


Access permissions

- What minimal permissions (minimal sum of permission weights) must be set on files or directories, such that you can successfully execute the following commands?

- `ls -ld ~/ps1-09/A`

~/	--x	---	---
~/ps1-09	--x	---	---
~/ps1-09/A	---	---	---

- `ls ~/ps1-09/A`

~/	--x	---	---
~/ps1-09	--x	---	---
~/ps1-09/A	r--	---	---

- `ls -l ~/ps1-09/A`

~/	--x	---	---
~/ps1-09	--x	---	---
~/ps1-09/A	r-x	---	---

Access permissions

- `ls -l ~/ps1-09/A/passwd`

```
~/                               --x  ---  ---
~/ps1-09                         --x  ---  ---
~/ps1-09/A                       --x  ---  ---
~/ps1-09/A/passwd                ---  ---  ---
```

- `cat ~/ps1-09/A/passwd`

```
~/                               --x  ---  ---
~/ps1-09                         --x  ---  ---
~/ps1-09/A                       --x  ---  ---
~/ps1-09/A/passwd                r--  ---  ---
```

- `echo "aaaaa" >> ~/ps1-09/A/passwd`

```
~/                               --x  ---  ---
~/ps1-09                         --x  ---  ---
~/ps1-09/A                       --x  ---  ---
~/ps1-09/A/passwd                -w-  ---  ---
```

Access permissions

- `echo "bbbbbb" > ~/ps1-09/A/passwd`

~/	--x	---	---
~/ps1-09	--x	---	---
~/ps1-09/A	--x	---	---
~/ps1-09/A/passwd	-w-	---	---

- `~/ps1-09/A/date`

~/	--x	---	---
~/ps1-09	--x	---	---
~/ps1-09/A	--x	---	---
~/ps1-09/A/date	--x	---	---

- `rm ~/ps1-09/A/passwd`

~/	--x	---	---
~/ps1-09	--x	---	---
~/ps1-09/A	-wx	---	---
~/ps1-09/A/passwd	---	---	---

Default access permissions

- What access permissions will have new created directory and file?
Why?

```
umask -S
```

```
umask
```

- What must be done to newly created files/directories should have automatically the following access rights (don't use command `chmod`)?

- directory: `rw- --- ---` file: `rw- --- ---`

```
umask -S u=rwx,g=,o=
```

```
umask 077
```

- directory: `rw- -w- r--` file: `rw- -w- r--`

```
umask -S u=rw,g=w,o=r
```

```
umask 153
```



Command find

- How to print only the number of regular files, which are in the directory /usr/bin (recursively) on the standard output?

```
find /usr/bin -type f 2>/dev/null | wc -l
```

- How to print only the number of symbolic links, which are in the directory /usr/bin (recursively) on the standard output?

```
find /usr/bin -type l 2>/dev/null | wc -l
```

- How to print only the number of regular files and symbolic links, which are in the directory /usr/bin (recursively) on the standard output?

```
find /usr/bin \( -type f -o -type l \) \  
2>/dev/null | wc -l
```

Command find

- Create files and directory by the following commands.

```
mkdir -p A/B/C
```

```
touch {A,A/B,A/B/C}/\
{,a,b,c}{,k,l,m}{,x,y,z}.{c,cpp,tar,gz,txt}
```

- ❶ How to print names of regular files, which have a suffix of length 3. (eg. abc.txt or xz.cpp)?

```
find . -type f -name "*.???"
```

- ❷ How to print names of regular files, which consist of a prefix of length 2 and the suffix .c or .cpp (eg. ab.c or xz.cpp)?

```
find . -type f \( -name '??.' -o -name '?.cpp' \)
```

- ❸ How to remove files found in question 1?

```
find . -type f -name "*.???" -ok rm {} \;
```

```
find . -type f -name "*.???" -exec rm {} \;
```

Command find

- How to print names of regular files from your home directory (recursively), that were modified during today, and how to verify the result by the command stat?

```
find . -type f -mtime 0 \  
-exec stat --printf="%n\t%y\n" {} \;
```

- How to print names of regular files from your home directory (recursively), that were modified during last 3 days, and how to verify the result by the command stat?

```
find ~ -type f -mtime -2 \  
-exec stat --printf="%y %n\n" {} \;
```

- How to print only names of all regular files, which have set write permission for owner or exec permission for other and are located in the directory /etc (recursively)? For every such file run command `ls -l` to verify the permissions.

```
find /etc -type f \( -perm -200 -o -perm -001 \) \  
-exec ls -l {} \; 2>/dev/null
```

Command find

- How to print only names of regular files, which are shell scripts and are located in the directory /usr/bin (recursively), on the standard output?

- ① Hint: The script is the file with the following first line

```
#!/bin/sh
```

```
find /usr/bin -type f \  
-exec grep -n "^#! */bin/*.sh" {} /dev/null \; | \  
grep "[^:]*:1:" | cut -d: -f1
```

- ② Hint: Use the command file.

```
find /usr/bin -type f -exec file {} \; 2>/dev/null | \  
grep script
```

- How to print all hard links of the file /etc/init.d/pppd in directory /etc (recursively) on the serve fray1.fit.cvut.cz.

```
find /etc \  
-inum $(ls -i /etc/init.d/pppd | awk '{print $1}')
```

```
-ls 2>/dev/null
```


Homework

- How to print names of regular files from your home directory, that have size bigger then 2 megabytes, and how to verify the result by the command `stat`?
- How to print names of all files from your home directory, that were accessed 7 days ago, and how to verify the result by the command `stat`?
- How to print names of regular files from the directory `/tmp`, that you can read, and how to verify the result by the command `ls`?