# Unix-like Operating Systems

## Access permissions and command find.

Jan Trdlička

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

# Contents

# User accout

- 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 "^ *$$"
```

```
pcred $$                # in Solaris
```

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

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

```
pcred $$                # 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 the directory `/tmp` 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 uos
dr-x------ user group uos/A
-r-x------ user group uos/A/date
-r-x------ user group uos/A/passwd
```

```
cd /tmp
mkdir -p uos/A

cp /etc/passwd uos/A
cp /bin/date uos/A

chmod -R 500 uos
```

# 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 /tmp/uos/A`

    ```
    /tmp            rwx rwx rwt
    /tmp/uos        --x --- ---
    /tmp/uos/A      --- --- ---
    ```

  - `ls /tmp/uos/A`

    ```
    /tmp            rwx rwx rwt
    /tmp/uos        --x --- ---
    /tmp/uos/A      r-- --- ---
    ```

  - `ls -l /tmp/uos/A`

    ```
    /tmp            rwx rwx rwt
    /tmp/uos        --x --- ---
    /tmp/uos/A      r-x --- ---
    ```

# Access permissions

- `ls -l /tmp/uos/A/passwd`

```
/tmp                    rwx rwx rwt
/tmp/uos                --x --- ---
/tmp/uos/A              --x --- ---
/tmp/uos/A/passwd       --- --- ---
```

- `cat /tmp/uos/A/passwd`

```
/tmp                    rwx rwx rwt
/tmp/uos                --x --- ---
/tmp/uos/A              --x --- ---
/tmp/uos/A/passwd       r-- --- ---
```

- `echo "aaaaa" >> /tmp/uos/A/passwd`

```
/tmp                    rwx rwx rwt
/tmp/uos                --x --- ---
/tmp/uos/A              --x --- ---
/tmp/uos/A/passwd       -w- --- ---
```

# Access permissions

- `echo "bbbbb" > /tmp/uos/A/passwd`

```
/tmp                    rwx rwx rwt
/tmp/uos                --x --- ---
/tmp/uos/A              --x --- ---
/tmp/uos/A/passwd       -w- --- ---
```

- `/tmp/uos/A/date`

```
/tmp                    rwx rwx rwt
/tmp/uos                --x --- ---
/tmp/uos/A              --x --- ---
/tmp/uos/A/date         --x --- ---
```

- `rm /tmp/uos/A/passwd`

```
/tmp                    rwx rwx rwt
/tmp/uos                --x --- ---
/tmp/uos/A              -wx --- ---
/tmp/uos/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: rwx --- ---  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}
```

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

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

2. 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 \( -name '??.c' -o -name '??.cpp' \)
```

3. 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`?