Διάλεξη 3

Διαχείριση Συστημάτων UNIX I

Δημήτρης Ζεϊναλιπούρ

Περιεχόμενο Διάλεξης

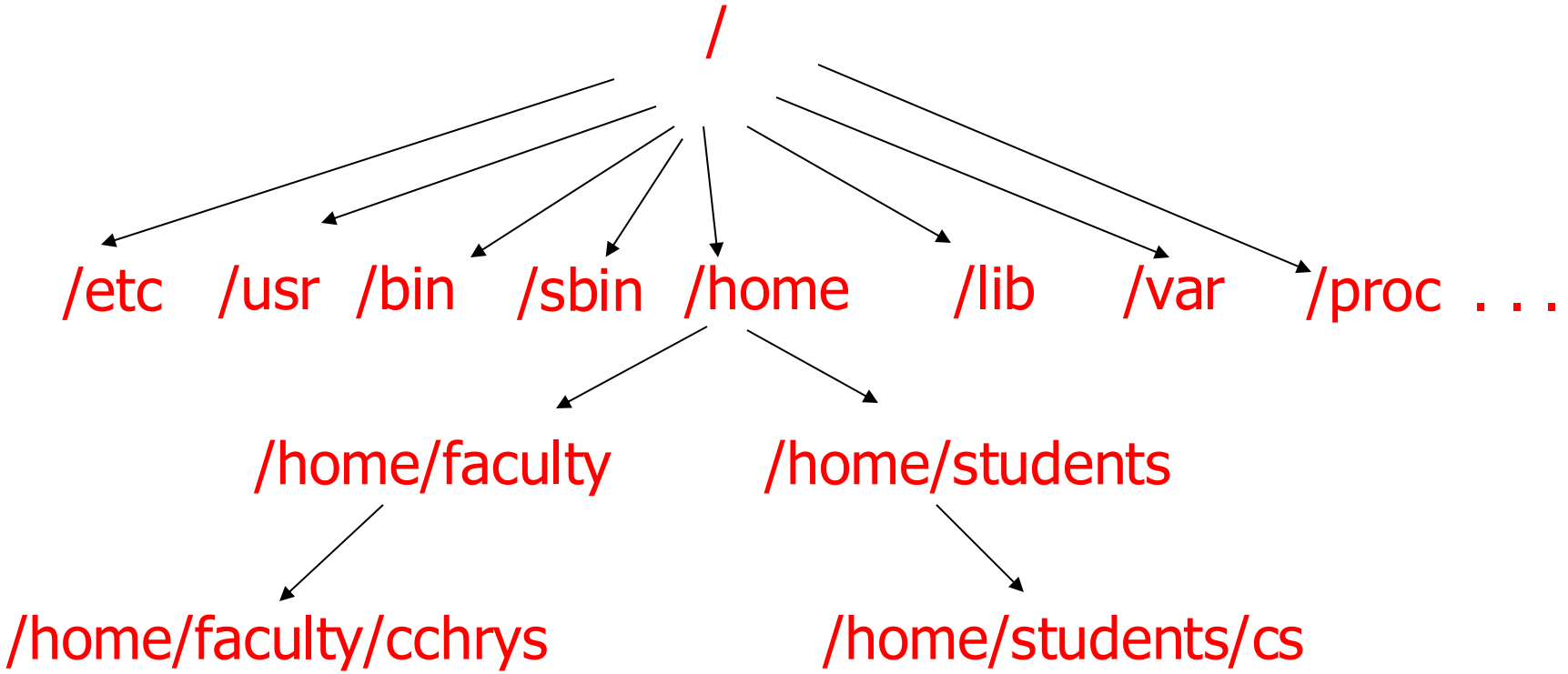
- ***Δομή καταλόγων*** *(ls, cd, pwd, pathnames, pushd, popd), χώρος δίσκου (du, df), Σύνδεσμοι (συμβολικοί, σκληροί - ln).*
- ***Κατάστιχα (/var/log)*** *και* ***Partitions*** *(mounts atime/otime/mtime, lsblk, mkfs, fdisk)*
- ***Συμπίεση-αποσυμπίεση*** *(zip/unzip, gzip/gunzip, bzip2/bunzip2, tar)*
  - *Παράδειγμα Backup (με public/private keys)*
- ***Ιδιοκτησία και Δικαιώματα Πρόσβασης*** *(chmod, chgrp, chown, umask, suid, sgid, sticky bit),*
- ***Προσθήκη Χρηστών*** *(useradd/ldap,getent)*

Δομή Καταλόγων UNIX

- Το **UNIX μεταχειρίζεται** τα πάντα ως **αρχεία**
  - Standard text files and binaries
  - Κατάλογοι (Directories)
  - Σύνδεσμοι (Links)
  - Συσκευές
    - τερματικό (terminal), πληκτρολόγιο, σκληρός δίσκος, ..
- Αυτό πάει πίσω στο σχεδιασμό του UNIX **προσπαθώντας** να κρατήσει τα **πάντα απλά**.
  - Μεταχειρίζοντας τα πάντα το ίδιο επιτρέπει σε μια απλή διεπαφή να αλληλεπιδρά με όλα με τον ίδιο τρόπο.

Δομή Καταλόγων UNIX

- Δομή Αντεστραμμένου Δέντρου



Δομή Καταλόγων UNIX

- **/**
  - Ριζικός κατάλογος (Root) ολόκληρου του συστήματος
- **/boot**
  - Αρχεία που χρησιμοποιούνται κατά την εκκίνηση.
  - Το “/boot/vmlinuz...” μονολιθικός πυρήνας του Linux ή /mach_kernel πυρήνας του MACOSX;
- **/bin**
  - Binaries (τα σημαντικότερα, όπως shells, ls, grep, …)
- **/usr/bin**
  - Binaries (λιγότερο σημαντικά, εργαλεία που χρησιμοποιούνται από τον χρήστη όπως mysql, gnome, python, zip, κτλ …)

Δομή Καταλόγων UNIX

- **/sbin και /usr/sbin**
  - System Binaries (Εκτελέσιμα που χρησιμοποιούνται για συντήρηση
    - Χαμηλού Επιπέδου (/sbin): reboot, grub (bootloader), mount, tcpdump, iptables, nmap, κτλ.
    - Υψηλού Επίπεδου (/usr/bin): π.χ., httpd, squid, sshd, vsftpd, dovecot κτλ.
- **/usr**
  - Περιέχει δεδομένα που σχετίζονται με τους χρήστες, π.χ.,
    - user binaries, τα έγγραφά τους, βιβλιοθήκες, επικεφαλίδες αρχείων, κλπ…

Δομή Καταλόγων UNIX

- **/lib (MacOSX => usr/lib & /Library)**
  - **Διαμοιραζόμενες Βιβλιοθήκες (Shared libraries)** για προγράμματα που γίνονται linked δυναμικά (όπως τα DLL's στα Windows)
  - Π.χ., /lib/libc-2.5.so είναι η βιβλιοθήκη της C στο UNIX η οποία δένεται δυναμικά με το εκτελέσιμο. Στα windows από την άλλη προγράμματα C είναι statically linked με τις βιβλιοθήκες (παρεχεται από τον compiler)
- **/dev**
  - Συσκευές (π.χ. disk, cdrom, dvd, port, audio, κλπ). Θυμηθείτε ότι στο Unix τα πάντα είναι αρχεία!
- **/home (MacOSX => /Users)**
  - Αποθηκεύονται οι λογαριασμοί χρηστών
  - Π.χ., /home/facultly/dzeina/

Δομή Καταλόγων UNIX

- **/var**
  - Χώρος όπου διατηρούνται δεδομένα τα οποία αλλάζουν συχνά (π.χ., logs, emails, εργασίες που γίνονται queued, π.χ., printer jobs)
- **/proc (not available on MacOSX ☹)**
  - Χώρος όπου «φυλάγονται» (με νοητό τρόπο όχι πραγματικά) πληροφορίες για τις υπό εκτέλεση διεργασίες (ανοικτά αρχεία, μνήμη, κτλ)
    - Π.χ., cat /proc/$$/status δείχνει πληροφορίες για την διεργασία του κελύφους που εκτελείτε.
- **/etc:** Configuration files (inittab: processes started at system bootup, fstab: file systems and their mount points)
- **/lost+found:** files that it restores after a system crash

Δομή Καταλόγων στο Android (Linux Kernel 2.6)

SH0APPL00803:
app-cache
config
cache
sdcard
acct
mnt
d
etc
system
sys
shutdown.bravo.rc
sbin
proc
init.rc
init.goldfish.rc
init.bravo.rc
init
default.prop
data
bootcomplete.bravo.rc
root
dev

Simple:
Use terminal app from Play Store (e.g., iTerminal

Programmer:
ADB Bridge part of the android SDK
Automations!
$ adb shell ls -R (list subdirectories recursively)
https://gist.github.com/Pulimet/5013acf2cd5b28e55036c82c91bd56d8

Researcher:
[C51] "Managing Smartphone Testbeds with SmartLab", Georgios Larkou, Constantinos Costa, Panayiotis Andreou, Andreas Konstantinidis, Demetrios Zeinalipour-Yazti, **Proceedings of the 27th USENIX Large Installation System Administration Conference** (**LISA'13**), Washington D.C., USA pp. 115-132, ISBN: 978-1-931971-05-8, **2013**.

Κατάστιχα /var/log

- Some of the most important Linux system logs include:
  - **/var/log/syslog (Debian)** and **/var/log/messages (**RHEL or CentOS) store all global system activity data, including startup messages.
  - **/var/log/boot.log:** Boot sequence log messages.
  - **/var/log/auth.log (Ubuntu and Debian)** and **/var/log/secure** (Red Hat and CentOS) store all security-related events such as logins, root user actions, and output from pluggable authentication modules (PAM).
  - **/var/log/kern.log** stores kernel events, errors, and warning logs, which are particularly helpful for troubleshooting custom kernels.
  - **/var/log/cron** stores information about scheduled tasks (cron jobs). Use this data to verify your cron jobs are running successfully.
  - **/var/log/apache2 *(Ubuntu)** or **/var/log/httpd (**CentOS, Fedora, RHEL)

Example /var/log/syslog

sudo cat /var/log/syslog

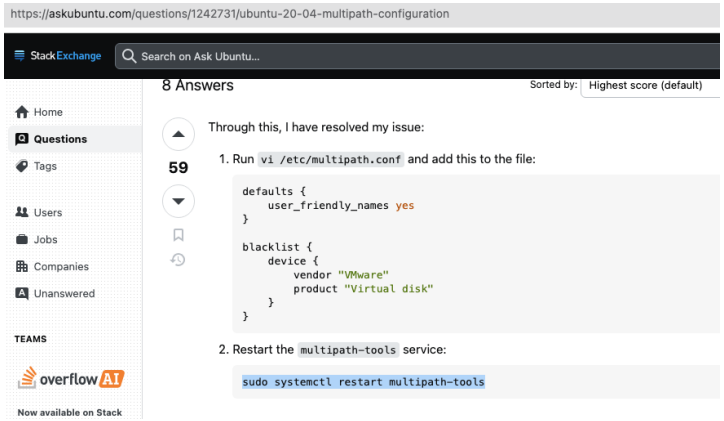
- Oct 22 11:08:35 vgate multipathd[764]: sda: add missing path
- **Oct 22 11:08:35 vgate multipathd[764]: sda: failed to get udev uid: Invalid argument**
- **Oct 22 11:08:35 vgate multipathd[764]: sda: failed to get sysfs uid: Invalid argument**
- **Oct 22 11:08:35 vgate multipathd[764]: sda: failed to get sgio uid: No such file or directory**
- Oct 22 11:08:37 vgate CRON[1294788]: (CRON) info (No MTA installed, discarding output)
- Oct 22 11:08:37 vgate CRON[1294787]: (CRON) info (No MTA installed, discarding output)
- Oct 22 11:08:40 vgate multipathd[764]: sda: add missing path
- **Oct 22 11:08:40 vgate multipathd[764]: sda: failed to get udev uid: Invalid argument**
- **Oct 22 11:08:40 vgate multipathd[764]: sda: failed to get sysfs uid: Invalid argument**
- **Oct 22 11:08:40 vgate multipathd[764]: sda: failed to get sgio uid: No such file or directory**
- Oct 22 11:08:45 vgate multipathd[764]: sda: add missing path
- **Oct 22 11:08:45 vgate multipathd[764]: sda: failed to get udev uid: Invalid argument**
- **Oct 22 11:08:45 vgate multipathd[764]: sda: failed to get sysfs uid: Invalid argument**
- **Oct 22 11:08:45 vgate multipathd[764]: sda: failed to get sgio uid: No such file or directory**
- Oct 22 11:08:50 vgate multipathd[764]: sda: add missing path
- **Oct 22 11:08:50 vgate multipathd[764]: sda: failed to get udev uid: Invalid argument**
- **Oct 22 11:08:50 vgate multipathd[764]: sda: failed to get sysfs uid: Invalid argument**
- **Oct 22 11:08:50 vgate multipathd[764]: sda: failed to get sgio uid: No such file or directory**
- Oct 22 11:08:55 vgate multipathd[764]: sda: add missing path

1) Observer Error

2) Find Solution

3) Monitor Situtat

4) Log files are automatically maintained/purge



Apache Logs /var/log/apache2

- The usual format that Apache follows for presenting log entries is:

```
"%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\""
```

- Here's how to interpret this formatting:
  - **%h** – The IP address of the client.
  - **%l** – This is the 'identd' on the client, which is used to identify them. This field is usually empty, and presented as a hyphen.
  - **%u** – The user ID of the client, if HTTP authentication was used. If not, the log entry won't show anything for this field.
  - **%t** – Timestamp of the log entry.
  - **\%r** – The request line from the client. This will show what HTTP method was used (such as GET or POST), what file was requested, and what HTTP protocol was used.
  - **%>s** – The status code that was returned to the client. Codes of 4xx (such as 404, page not found) indicate client errors and codes of 5xx (such as 500, internal server error) indicate server errors. Other numbers should indicate success (such as 200, OK) or something else like redirection (such as 301, permanently moved).
  - **%O** – The size of the file (including headers), in bytes, that was requested.
  - **\"%{Referer}i\"** – The referring link, if applicable. This tells you how the user navigated to your page (either from an internal or external link).
  - **\"%{User-Agent}i\"** – This contains information about the connecting client's web browser and operating system.
- A typical entry in the access log will look something like this:
  - 10.10.220.3 - - [17/Dec/2019:23:05:32 -0500] "GET /products/index.php HTTP/1.1" 200 5015 "http://example.com/products/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.79 Safari/537.36"

Access Logs /var/log/wtmp

- **last** searches back through the /var/log/wtmp file (or the file designated by the **-f** option) and displays a list of all users logged in (and out) since that file was created.

```
vgate@vgate:~$ last | head
vgate    pts/0        10.16.16.23     Tue Oct 22 11:08   still logged in
vgate    pts/0        10.16.16.43     Tue Oct 22 11:08 - 11:08  (00:00)
vgate    pts/0        10.16.16.13     Mon Oct 21 10:13 - 10:15  (00:02)
vgate    pts/0        10.16.16.13     Mon Oct 21 09:04 - 09:04  (00:00)
vgate    pts/0        10.16.16.13     Mon Oct 21 08:52 - 08:53  (00:00)
vgate    pts/0        10.16.16.13     Mon Oct 21 08:51 - 08:52  (00:00)
vgate    pts/0        10.16.16.13     Mon Oct 21 06:00 - 06:02  (00:01)
vgate    pts/0        10.16.16.13     Mon Oct 21 05:58 - 06:00  (00:01)
vgate    pts/0        10.16.16.13     Thu Oct 17 09:47 - 12:00  (02:13)
vgate    pts/0        10.16.16.13     Thu Oct 17 08:38 - 08:41  (00:02)
```

Κατάστιχα systemd / journalctl

- The **`journalctl`** command is part of the **systemd** suite of utilities and is used to query and display log messages from the systemd journal. (in binary form)
  - The **systemd journal** and **/var/log/syslog** are two different logging mechanisms in Linux, with the systemd journal being the default for systemd-managed systems, while /var/log/syslog has been a standard for Unix-like systems for decades.
- The **systemd journal** is a centralized logging system that collects and stores log data from various sources, including system services, kernel events, and user applications **(in binary form)**
- /var/log/apache2 vs. systemd / journalctl
  - Manages system services and **processes during boot and runtime**. It handles starting, stopping, and supervising services like Apache (but Apache individual log files are more complete if we are interested in understanding its internal operation)

Πλοήγηση στη Δομή Καταλόγων

- Εντολή *ls* (επιλογές *-a, -l, -r, -R, -t, -i*)
  - Λίστα αρχείων και καταλόγων στο υφιστάμενο κατάλογο

-a : (all) do not ignore files starting with . (hidden files)

-l : use a long listing format

-rw-r--r-- ***1*** *dzeina* ***faculty*** *950784* ***Sep 14 13:29*** *01.Introduction.ppt*

-r : reverse order while sorting (αντίστροφη ταξινόμηση)

-R : list subdirectories recursively (αναδρομικά)

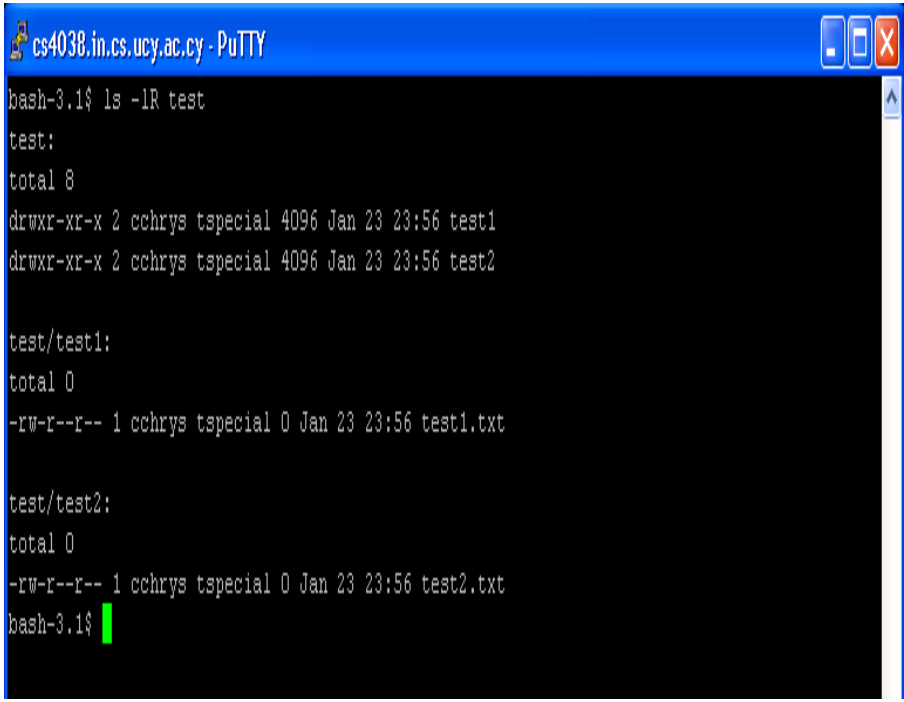
-t : sort by modification time (ταξινόμηση βάσει του χρόνου τροποποίησης)

-i : print the index (inode) number of each file

1758427225 *drwxr-x--- 2 dzeina faculty 40 Oct 24 2006 zei*

Πλοήγηση στη Δομή Καταλόγων

- Ιδιότητες Αρχείων
  - *Δικαιώματα*
  - *Αριθμό Σκληρών Συνδέσμων*
  - *Ιδιοκτήτης*
  - *Ομάδα Ιδιοκτήτη*
  - *Μέγεθος*
  - *Χρόνος Τροποποίησης*
  - *Όνομα*



drwxr-x--- 2 dzeina faculty 40 Oct 24 2006 zei

File Creation Timestamps
[atime, mtime, ctime and crtime/btime (ext4)]

```
$ touch test.txt
$ stat test.txt
  File: 'test.txt'
  Size: 0         Blocks: 0          IO Block: 1048576 regular empty file
Device: 3fh/63d Inode: 6670939444  Links: 1
Access: (0600/-rw-------)  Uid: ( 1240/  dzeina)   Gid: (  231/ faculty)
Context: system_u:object_r:nfs_t:s0
Access: 2018-03-28 11:31:19.705349937 +0300   # time of last access (of the file's content) – R *
Modify: 2018-03-28 11:31:19.705349937 +0300   # time of last data modification (file's content) – W or A
Change: 2018-03-28 11:31:19.705349937 +0300   # time of status change (inode change)
Birth: -  # ext4 newly introduced attribute to show when it appeared on filesystem
$ echo "a" > test.txt
$ stat test.txt
  File: 'test.txt'
  Size: 2         Blocks: 8          IO Block: 1048576 regular file
Device: 3fh/63d Inode: 6670939444  Links: 1
Access: (0600/-rw-------)  Uid: ( 1240/  dzeina)   Gid: (  231/ faculty)
Context: system_u:object_r:nfs_t:s0
Access: 2018-03-28 11:31:19.705349937 +0300
Modify: 2018-03-28 11:31:40.629763232 +0300
Change: 2018-03-28 11:31:40.629763232 +0300
 Birth: -
```

* Relatime (deferred atime updates) in 2 slides

File Creation Timestamps [atime, mtime, ctime and crtime/btime (ext4)]

```
$ chmod 777 test.txt
$ stat test.txt
  File: 'test.txt'
  Size: 2         Blocks: 8          IO Block: 1048576 regular file
Device: 3fh/63d Inode: 6670939444  Links: 1
Access: (0777/-rwxrwxrwx)  Uid: ( 1240/  dzeina)   Gid: (  231/ faculty)
Context: system_u:object_r:nfs_t:s0
Access: 2018-03-28 11:31:19.705349937 +0300 # time of last access (of the file's content) – Read
Modify: 2018-03-28 11:31:40.629763232 +0300 # time of last data modification (file's content) – Write or A
Change: 2018-03-28 11:33:14.851150111 +0300 # time of status change (inode change)
Birth: -
$ cat test.txt
a
$ stat test.txt
  File: 'test.txt'
  Size: 2         Blocks: 8          IO Block: 1048576 regular file
Device: 3fh/63d Inode: 6670939444  Links: 1
Access: (0777/-rwxrwxrwx)  Uid: ( 1240/  dzeina)   Gid: (  231/ faculty)
Context: system_u:object_r:nfs_t:s0
Access: 2018-03-28 11:33:32.247672734 +0300
Modify: 2018-03-28 11:31:40.629763232 +0300
Change: 2018-03-28 11:33:14.851150111 +0300
Birth: -
```

Noatime, Relatime mounts

- **Normal mounting according to POSIX is atime, however certain filesystems use noatime (atime never accessed) or relatime**
- **relatime** maintains atime data, but not for each time that a file is accessed.
  - With this option enabled, atime data is written to the disk only if:
    - the file has been modified since the atime data was last updated (mtime - time of last **file's content** data **m**odification – W or
    - the file was last accessed more than a certain amount of **time ago (by default, one day).**

```
cat /proc/mounts | grep students
pallene.in.cs.ucy.ac.cy:/home/students /home/students nfs4
rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,
proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=10.16.1.101,loca
l_lock=none,addr=10.16.1.33 0 0
```

Πλοήγηση στη Δομή Καταλόγων

- Εντολή *cd*
  - Αλλαγή τρέχοντος καταλόγου

- Ειδικοί Κατάλογοι-Συμβολισμοί
  - Κατάλογος-ρίζα (/)
  - Τρέχων κατάλογος (.)
  - Γονικός κατάλογος (..)
  - Κατάλογος Αφετηρίας ($HOME) (~)
  - Αυτοί οι κατάλογοι μπορούν να «στοιβαχθούν»
    - Π.χ. ../.. ➔ δυο κατάλογοι πάνω από τον τρέχων κατάλογο

Πλοήγηση στη Δομή Καταλόγων

- Εντολή *pwd (ή echo $PWD)*
  - Εμφάνιση **Απόλυτου** ονόματος-μονοπατιού τρέχοντος καταλόγου
- Ονόματα-Μονοπάτια
  - **Απόλυτα**
    - Πάντοτε ξεκινά από τον κατάλογο-ρίζα (/)και περιλαμβάνει όλη τη διαδρομή
      - Π.χ. */home/faculty/dzeina/public*
  - **Σχετικά**
    - Σχετικά με τον τρέχων κατάλογο
      - Π.χ. *~/public*

Προχωρημένη Πλοήγηση στη Δομή Καταλόγων

- Εντολή *pushd* <*dir*>
  - «Σπρώχνει» ένα κατάλογο σε μια στοίβα
    - Επίσης μετακινούμαστε στο κατάλογο <dir>
- Εντολή *popd*
  - «Βγάζει» την κεφαλή X από τη στοίβα
    - Επίσης μετακινούμαστε στο κατάλογο <X>
- Μια στοίβα είναι γνωστή ως *LIFO*
  - *Last In, First Out*
- Εκτελώντας: *pushd*
  - παραθέτει το περιεχόμενο της στοίβας.

Χώρος Δίσκου και Όριο Χρήσης

- *Έλεγχος χρήσης χώρου δίσκου*
  - Εντολή *du* (disk usage)
    - Δείχνει πόσο χώρο (σε kilobytes) δεσμεύεται για κάθε αρχείο ή κατάλογο (**αναδρομικά**)
      - *du –c:* ***συνοψίζει το τελικό άθροισμα σε bytes.***
      - *du –s:* ***εκτελεί την ανάδρομη αλλά παρουσιάζει*** *μόνο το άθροισμα του μεγέθους του καταλόγου (όχι το μέγεθος κάθε επί μέρους καταλόγου)*

Παραδείγματα

du -c answers.txt input.txt

28 answers.txt

4 input.txt

32 total

du zei

0 zei/s1

20 zei

du -s zei

20 zei

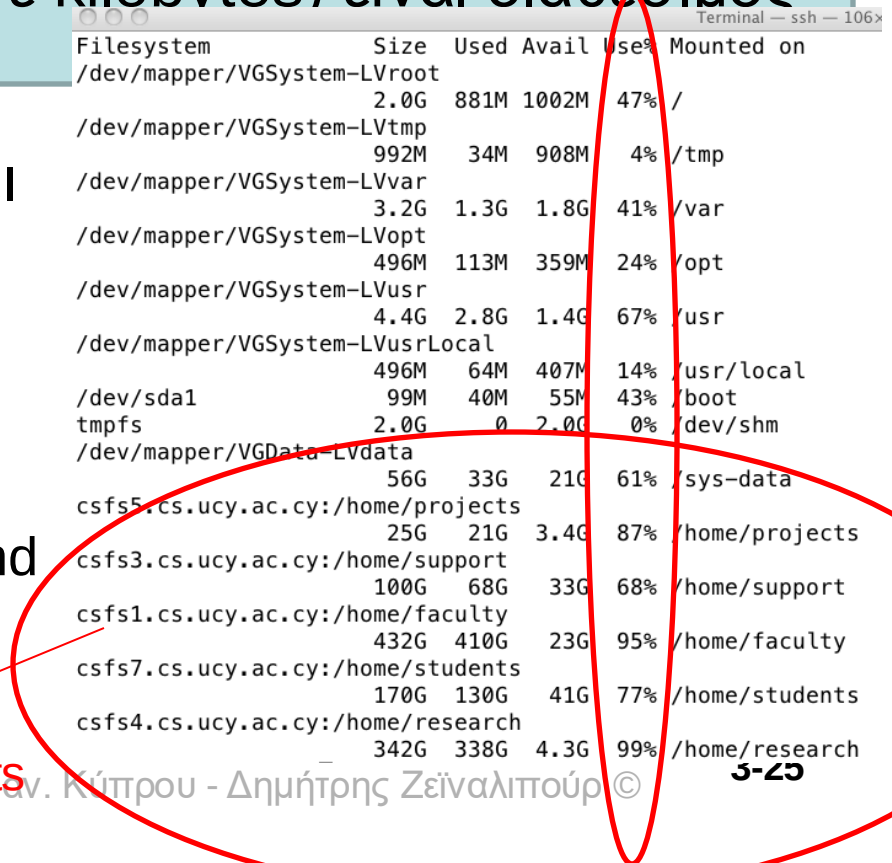
Find Largest File in UNIX (e.g., partition filled)

```
du –a /dir/ | sort –n –r | head –n 20
```

- *Έλεγχος διαθέσιμου χώρου δίσκου*
  - Εντολή *df –h (human readable output)*
    - Δείχνει πόσος χώρος (σε kilobytes) είναι διαθέσιμος στο file system

Ο λογαριασμός του χρήστη έχει περιορισμένο χώρο. Χρησιμοποιείται ένα *όριο χρήσης* (*quota*) για να ελεγχθεί ο διαθέσιμος χώρος.

quota –v # display disk usage and limits



NFS Mounts

Mount 1TB NVMe on Ubuntu Host

 # create mount folder

sudo mkdir /media/nvme1TB

 # format file system

sudo mkfs.ext4 /dev/sdb

 # mount filesystem

sudo mount /dev/sdb /media/nvme1TB/

```
sudo lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL
NAME                      FSTYPE      SIZE MOUNTPOINT    LABEL
sda                                     1T
├─sda1                    vfat        1G /boot/efi
├─sda2                    ext4        2G /boot
└─sda3                    LVM2_member 1020.9G
  └─ubuntu--vg-ubuntu--lv ext4        100G /
sdb                       ext4        1T /media/nvme1TB
sr0                                  1024M
```

$ mount # not mounted
yet :-(

```
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=51508796k,nr_inodes=
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=10309016k,
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,r
/dev/mapper/ubuntu--vg-ubuntu--lv on / type ext4 (rw,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,re
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,n
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=32,pgrp=
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatim
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,relatime,page
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,rel
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatim
/dev/sda2 on /boot type ext4 (rw,relatime)
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0022,dmask=0022,code
ro)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,node
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=103090
```

cd /media/nvme1TB; touch file # create a file on new mount

lsblk – Filesystem Info

lsblk lists information about all available or the specified block devices. The **lsblk** command reads the **sysfs** filesystem and **udev db** to gather information.

- **-f, --fs**
  - Output info about filesystems.
- **-a, --all**
  - Also list empty devices and RAM disk devices.

Find All Partitions with lsblk

```
dmslv100@dmslv100:~$ sudo lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL
[sudo] password for dmslv100:
NAME                  FSTYPE      SIZE MOUNTPOIN LABEL
sda                                 1T
├─sda1                vfat          1G /boot/efi
├─sda2                ext4          2G /boot
└─sda3                LVM2_member 1020.9G
  └─ubuntu--vg-ubuntu--lv ext4      100G /
sdb                                 1T
sr0                              1024M

ap@ap:~$ lsblk -f
NAME   FSTYPE   LABEL UUID                                   FSAVAIL FSUSE% MOUNTPOINT
fd0
loop5  squashfs                                               0   100% /snap/lxd/29619
sda
├─sda1
└─sda2 ext4           fcef59e2-4d33-410f-8a7d-01a9eacf9c04   38.1G    47% /
sdb
└─sdb1 ext4           47ccec79-bb14-4922-8fc7-72cf97ddc225  401.4G    13% /data

ap-lb ~ $ lsblk -a
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
fd0      2:0    1    4K  0 disk
loop7    7:7    0        0 loop
sda      8:0    0   80G  0 disk
├─sda1   8:1    0    1M  0 part # not mounted
└─sda2   8:2    0   80G  0 part /
sdb      8:16   0   500G  0 disk
└─sdb1   8:17   0   500G  0 part /data
sr0     11:0    1  1024M  0 rom
```

Find Data Partitions with fdisk

$sudo fdisk –l # manipulate disk partition table

Disk /dev/sda: 1 TiB, 1099511627776 bytes, 2147483648 sectors

Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: CB2BC496-C1F9-4A8F-8C6A-3869306055F9

Device	Start	End	Sectors	Size Type
/dev/sda1	2048	2203647	2201600	1G EFI System
/dev/sda2	2203648	6397951	4194304	2G Linux filesystem
/dev/sda3	6397952	2147481599	2141083648	1020.9G Linux filesystem

Disk /dev/sdb: 1 TiB, 1099511627776 bytes, 2147483648 sectors

Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/ubuntu--vg-ubuntu--lv: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Φύλαξη Χώρου Δίσκου

- Συμπίεση αρχείων
  - Εντολή *zip*
    - Κάθε αρχείο που συμπιέζεται αντικαθίσταται με ένα αρχείο με προέκταση *.zip*
    - *Παράδειγμα:* ***zip file.zip ****
  - Εντολή *gzip*
    - **GNU εκδοχή του *zip***
    - Κάθε αρχείο που συμπιέζεται αντικαθίσταται με ένα αρχείο με προέκταση *.gz*
  - Εντολή *bzip2*
    - Διαφορετικός αλγόριθμος (LZW)
    - Κάθε αρχείο που συμπιέζεται αντικαθίσταται με ένα αρχείο με προέκταση *.bz2*

Επιλογή *-v*

- Verbose mode – δείχνει την αναλογία συμπίεσης για κάθε αρχείο που επεξεργάζεται
- *Τυπώνει:* total bytes=55, compressed=44 -> 20% savings

Φύλαξη Χώρου Δίσκου

- Αποσυμπίεση αρχείων
  - Εντολή *unzip*
  - Εντολή *gzip -d* (decompress) ή gunzip
  - Εντολή *bzip2* -d ή *bunzip2*

Φύλαξη Χώρου Δίσκου

Verbosely (δηλ., δείχνει τι συμπιέζεται στο tar file)

list contents

create file

extract

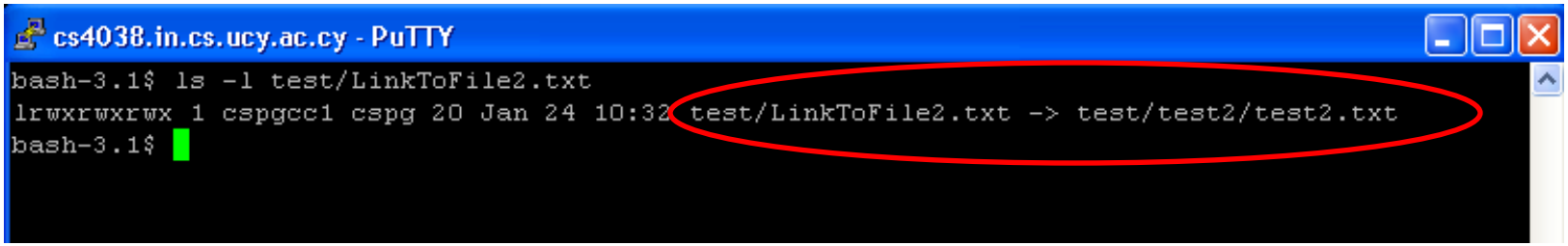
- Εντολή *tar* (επιλογές *-c, -f, -t, -v, -x*)
  - Σώζει πολλά αρχεία μαζί σ' ένα **ενιαίο αρχείο** και **διατηρεί πληροφορίες του filesystem όπως** user, group permissions, dates, directory structures.
  - TAR: Tape Archive
- *Παραδείγματα*
  - *tar -cvf* archive.tar foo bar
    - Δημιουργεί το *tar_file* από τα αρχεία foo και bar.
  - *tar -xvf* archive.tar
    - Εξάγει όλα τα αρχεία από το *tar_file*.
  - tar -cvfz archive.tar.gz foo bar
    - Δημιουργεί συμπιεσμένο *tar_file (με τον αλγοριθμο gzip)* από τα αρχεία foo και bar

Παράδειγμα Εργαλείου Backup

```
# Execute every day at 02:30
# crontab -e
# Add: 30 2 * * * /home/faculty/dzeina/backups/anyplace/wwwbackup.sh

#!/bin/bash
# ALWAYS remember to have #!/bin/bash (no spaces) in the first line of the script for
it to be executed by crond! Also check permissions to be 755 so that crond can have
execute permissions on the file
date

# carry out remote backup
echo -e "Backup WWW Play Web  ap.cs.ucy.ac.cy .."
ssh -l anyplace ap.cs.ucy.ac.cy tar -zcvf /tmp/anyplace-www-daily-backup.tar.gz /home/anyplace
echo "Done"


# transfer on csucy remote backup
echo -e "Transferring remote /tmp/anyplace-www-daily-backup.tar.gz locally ..."
scp anyplace@ap.cs.ucy.ac.cy:/tmp/anyplace-www-daily-backup.tar.gz
/home/faculty/dzeina/backups/anyplace/anyplace-www-daily-backup.tar.gz
echo "Done"


# delete remote backup
ssh -l anyplace ap.cs.ucy.ac.cy rm -rf /tmp/anyplace-www-daily-backup.tar.gz
```

Συμβολικοί Σύνδεσμοι

- Ειδικός τύπος αρχείου: ΌΧΙ αρχείο δεδομένων, αλλά αρχείο που περιέχει ένα *δείκτη* σε κάποιο άλλο αρχείο (δηλ., σαν Shortcut στα Windows).
  - Δρα ως *συντόμευση*
    - *Συντόμευση* σε ένα κατάλογο
    - Παρέχει ένα γρήγορο σύνδεσμο σε οποιοδήποτε αρχείο.

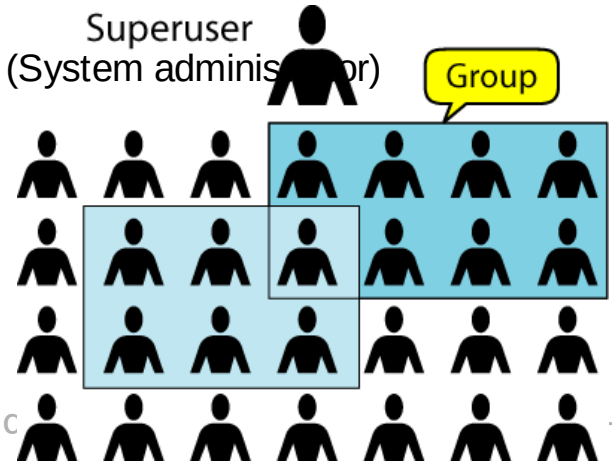


Σύνδεσμοι

- Εντολή *ln* (επιλογή *-s*) <*TARGET*> <*NAME*>
  - *-s* δηλώνει ένα ***συμβολικό* σύνδεσμο (symbolic link)**
  - χωρίς την επιλογή αυτή, ένας ***σκληρός* σύνδεσμος (hard link)** δημιουργείται με το περιεχόμενο του *target αρχείου*.
  - Π.χ., ln –s oldfile symboliclinkfile
- *Διαφορά μεταξύ ενός* σκληρού *και ενός συμβολικού* συνδέσμου:
  - Θα εξηγηθεί αργότερα πιο αναλυτικά.
  - Προκαταρτική επεξήγηση:
    - **Συμβολικός σύνδεσμος:** Ένα αρχείο (ή κατάλογος) Χ, μαζί με πολλές εγγραφές μετα-πληροφοριών (inodes) που αναφέρονται στο Χ.
      - Κάθε inode meta-πληροφοριών είναι σαν Shortcut
    - **Σκληρός σύνδεσμος:** Ένα αρχείο (ή κατάλογος) Χ, μαζί με 1 εγγραφή μέτα-πληροφοριών (inode) που αναφέρεται στο Χ. Η εγγραφή μέτα-πληροφοριών (inode) καταγράφει πόσοι αναφέρονται στο εν λόγω αρχείο.

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- ***Χρήστης (User)***: οποιοσδήποτε έχει λογαριασμό στο UNIX σύστημα
- Οι χρήστες οργανώνονται σε ***Ομάδες (Groups)***.
  - **ένας** ή **περισσότεροι** χρήστες μπορούν να ανήκουν σε **πολλαπλές ομάδες**.



Ιδιοκτησία και Δικαιώματα Πρόσβασης

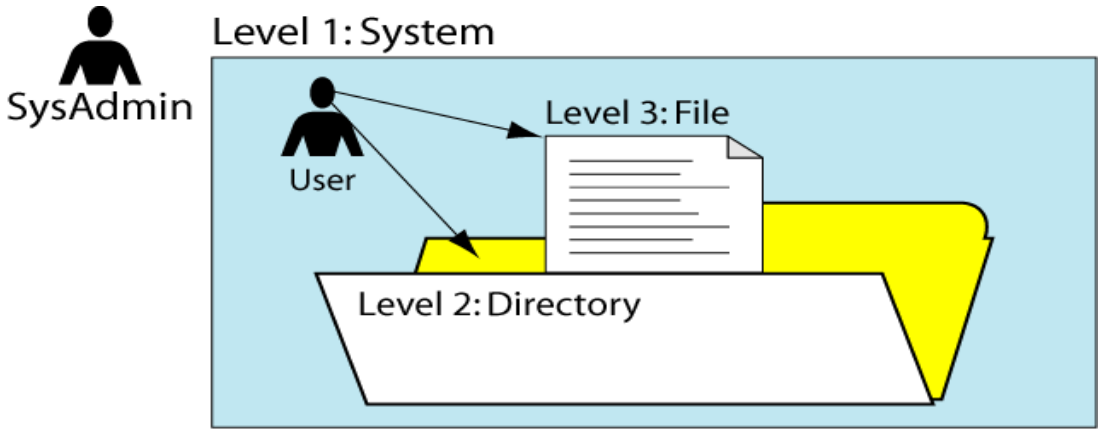
- Για να βρεις πληροφορίες σε ποια/ες ομάδα/ες ανήκει ένας χρήστης:
  - Εντολή *groups* <*username*>

```
bash-3.1$ groups dzeina
dzeina : faculty epl371
bash-3.1$ groups cchrys
cchrys : tspecial epl001 csphd visiting epl371
```

  - Σημείωση: Στο UNIX, κάθε χρήστης **ΠΡΕΠΕΙ** να είναι μέλος **τουλάχιστο μιας ομάδας** (αυτή που ορίζεται από το GID μέσα στο **/etc/passwd**)

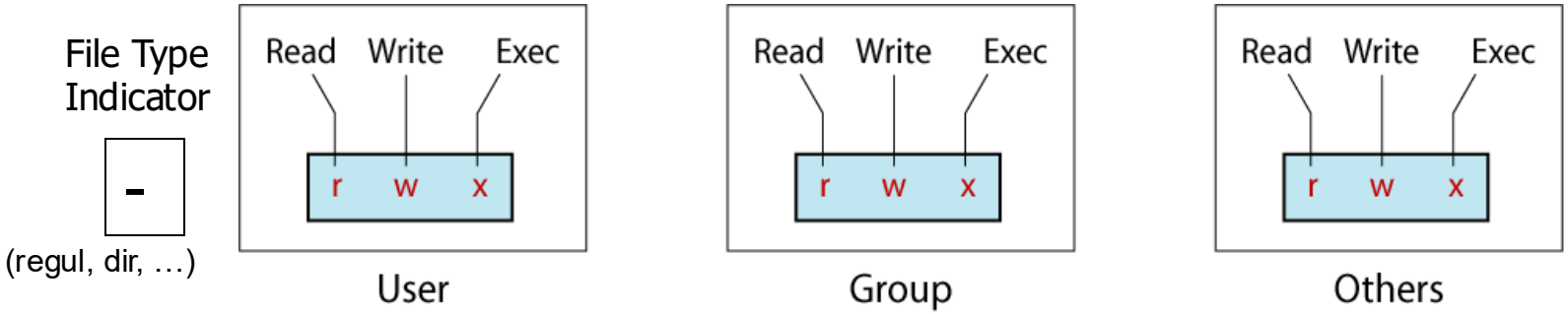
Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Επίπεδα Ασφάλειας:
  - **Σύστημα, Κατάλογος**, **Αρχείο**
  - **Ασφάλεια Συστήματος**: ελέγχεται **από τον διαχειριστή του συστήματος** (system administrator)
  - **Κατάλογος και Αρχείο**: ελέγχεται **από το χρήστη** στον οποίο ανήκει



Ιδιοκτησία και Δικαιώματα Πρόσβασης

- **Κώδικας Δικαιωμάτων Πρόσβασης***



  - **User (Χρήστης-Ιδιοκτήτης):** Ο δημιουργός του αρχείου
  - **Group (Ομάδα):** Σετ από χρήστες που ομαδοποιούνται
  - **Others (Υπόλοιποι):** Οποιοσδήποτε λογαριασμός που δεν ανήκει στην Ομάδα αλλά ανήκει σε άλλη ομάδα
  - Τρεις τύποι δικαιωμάτων πρόσβασης:
    - r read
    - w write
    - x execute
    - - permission denied

* ls -al <file | directory>

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Τι υποδηλώνει το κάθε είδος πρόσβασης;

Τύπος Πρόσβασης	Σημασία για Αρχείο	Σημασία για Κατάλογο
r (read)	View file contents	List directory contents
w (write)	Change file contents	- Change directory contents (create and remove files in that dir.)
x (execute)	Run executable file	- Access files explicitly (by name) in the given folder
-	Permission denied	Permission denied

How to provide RWX Access to specific Other?

- chmod xx7 folder
  - gives access rwx to other, might be necessary e.g., to allow the cron job user access a given directory.
  - At the same time other processes also have this right.
  - **How to fix it?**
- **Solution 1:** Make him the Owner of the folder
  - chown -R user2:group2 /home/user1/documents
  - Problem: we lose access to the folder (i.e., we become others :-(
- **Solution 2: setfacl/getfacl next slide …**

Access Control Lists (ACLs) of files and directories.

```
1)dzeina@ada> getfacl test/
# file: test/
# owner: dzeina
# group: faculty
user::rwx
group::---
other::---
```

- **setfacl**: Invokes the program to manage permissions.
- **-m**: This flag is used to Modify permissions.
- **u:root:rwx**: Specifies the user *root* (u:root) and grants him reading, writing, and execution permissions (:rwx).
- **test**: The target file/folder to which permissions are modified/applied.

```
1)dzeina@ada> setfacl -m u:root:rwx test
1)dzeina@ada> getfacl test/
# file: test/
# owner: dzeina
# group: faculty
user::rwx
user:root:rwx
group::---
mask::rwx
other::---
```

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Δείκτης Τύπου Αρχείου

Χαρακτήρας	Δείκτης Τύπου Αρχείου
d	**D**irectory
b	**B**lock-type special file (π.χ., DVD, CDROM, DISK)
c	**C**haracter-type special file (π.χ., terminals, printers και networks)
l	Symbolic **L**ink
p	**P**ipe
s	**S**ocket
-	**R**egular file

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Ελέγχοντας τα Δικαιώματα:
  - Εντολή *ls -l <file_OR_dir_name>*

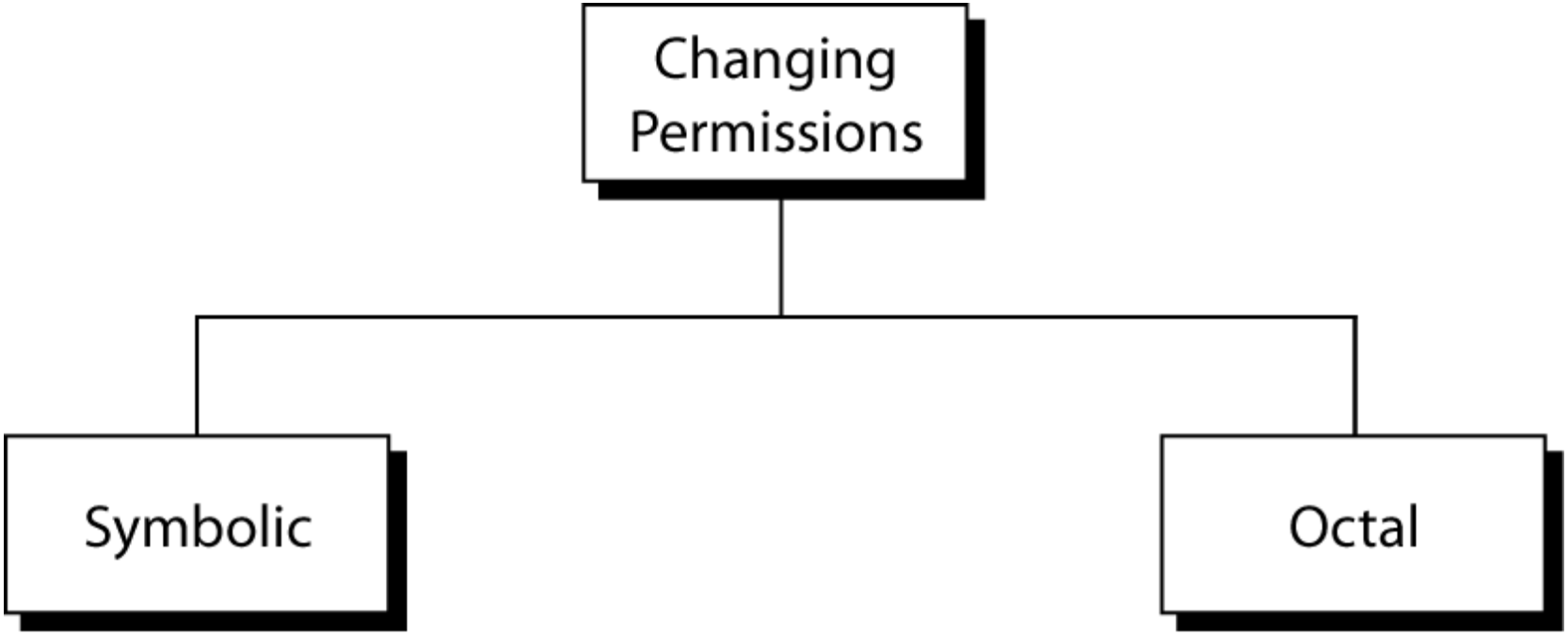
```
bash-3.1$ ls -lR test/ | head
test/:
total 4
-rw-r--r-- 2 cspgcc1 cspg 28 Jan 25 14:35 HardLinkToFile1.txt
lrwxrwxrwx 1 cspgcc1 cspg 20 Jan 24 10:50 SymbLinkToFile2.txt -> test/test2/test2.txt
drwxr-xr-x 4 cspgcc1 cspg 50 Jan 28 17:25 test1
drwxr-xr-x 2 cspgcc1 cspg 22 Jan 24 10:42 test2

test/test1:
total 4
drwxr-xr-x 2 cspgcc1 cspg 24 Jan 23 22:41 test1_1
```

total size of all files in the list (measured in 512 B)

Permissions, Hard-Links, User, Group, FileSize, Last Modified Date + Time, Filename

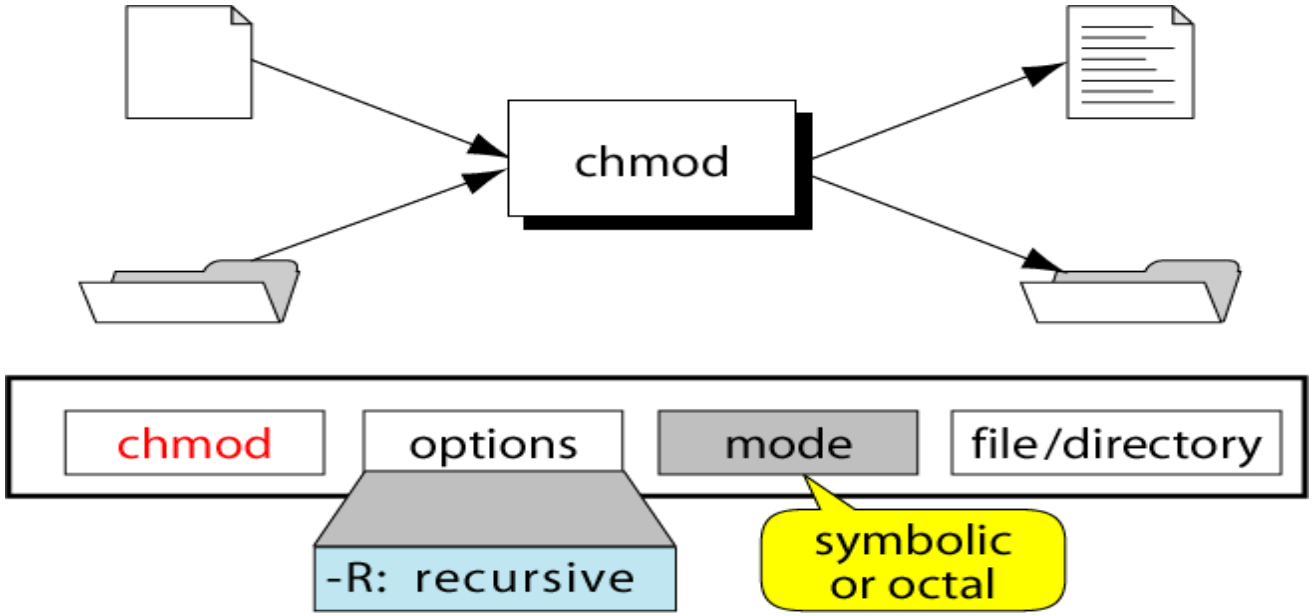
Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Αλλαγή Δικαιωμάτων:



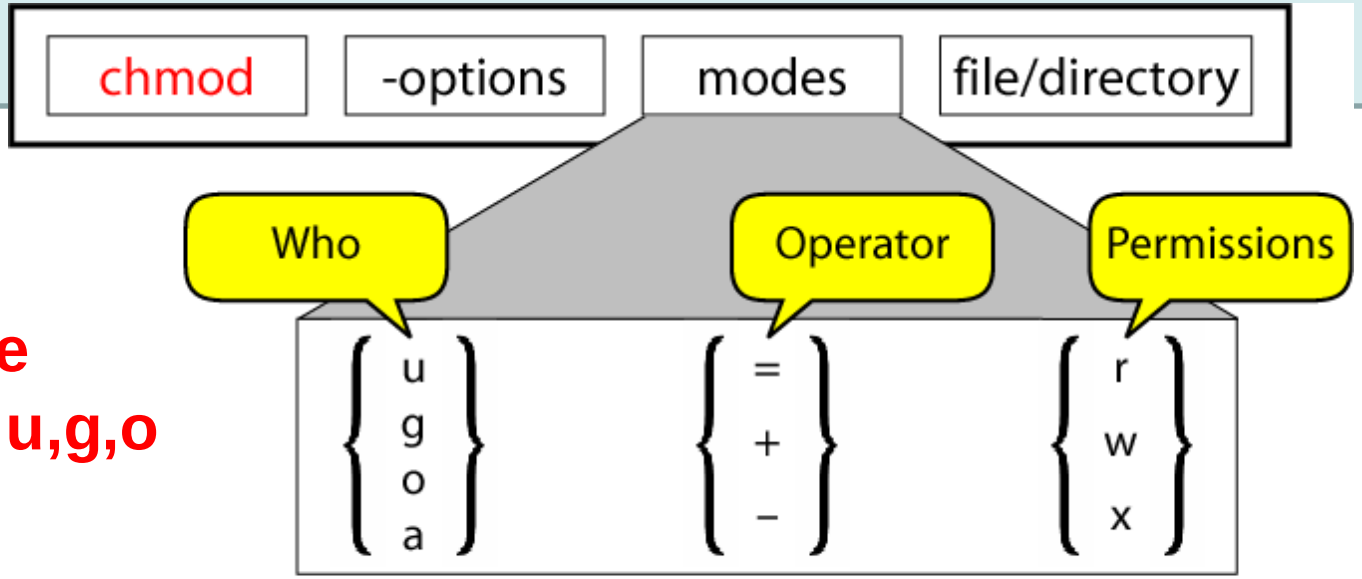
Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Αλλαγή Δικαιωμάτων:
  - Εντολή *chmod* → **μόνον ο ιδιοκτήτης (ο su)**
    (και άλλα μέλη της ομάδας κάτω από ορθά δικαιώματα μπορούν να το πράξουν!)
  -



Ιδιοκτησία και Δικαιώματα Πρόσβασης

- **Αλλαγή Δικαιωμάτων: Συμβολική μορφή**
  - Εντολή *chmod*



Chmod +x file
> Adds +x to u,g,o

Example

```
chmod   u=rwx,g+w,o-w   memo.doc
```

chmod +x

```
$touch testfile
$ls -al testfile
-rw-r--r-- 1 dzeina staff 0 Feb 14 11:27 testfile

$chmod +x testfile
$ls -al testfile
-rwxr-xr-x 1 dzeina staff 0 Feb 14 11:27 testfile

$chmod = testfile
$ ls -al testfile
---------- 1 dzeina staff 0 Feb 14 11:27 testfile

$chmod 777 testfile
$ ls -al testfile
-rwxrwxrwx 1 dzeina staff 0 Feb 14 11:27 testfile
```

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- **Αλλαγή Δικαιωμάτων: Συμβολική μορφή**
  - Εντολή *chmod*

chmod who operation permissions filename

u for user
g for group
o for others
a for all

\+ for add
\- for remove
= for assign (set)

r for read
w for write
x for execute

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Αλλαγή Δικαιωμάτων: **Συμβολική μορφή**
  - Παράδειγμα:

```
bash-3.1$ ls -l test.txt
-rw-r--r-- 1 cspgcc1 cspg 0 Jan 30 19:38 test.txt
```

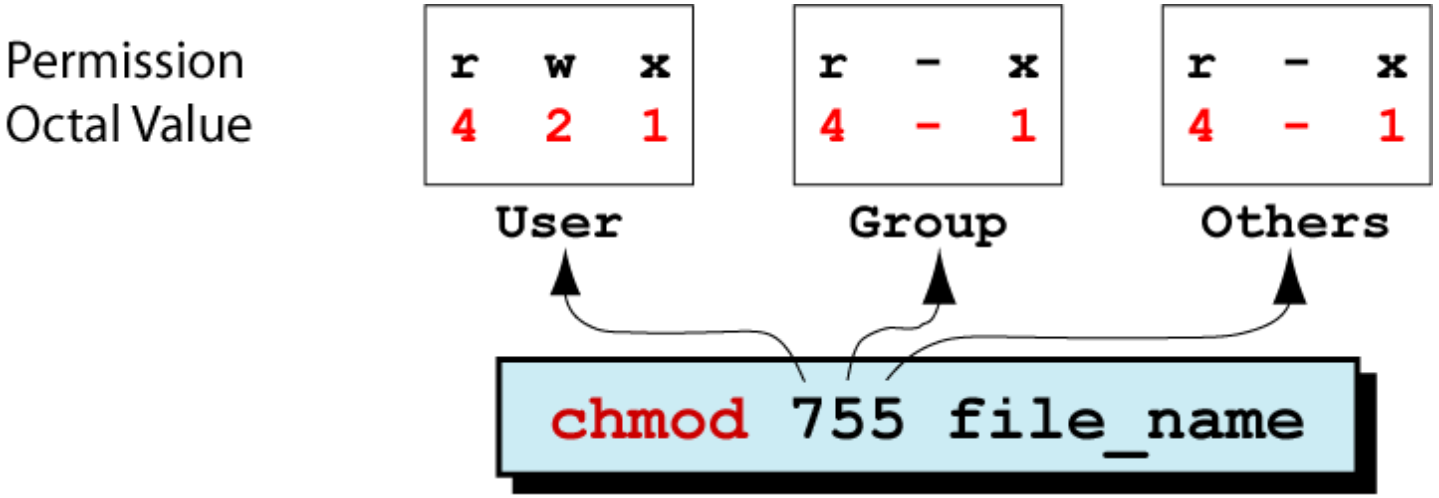
Αλλαγή των δικαιωμάτων πρόσβασης έτσι ώστε **όλοι να μπορούν να το διαβάζουν** και να το **εκτελούν** και **μόνο ο ιδιοκτήτης** και η **ομάδα** να **μπορούν να γράφουν σ' αυτό** (rwx|rwx|r-x):

```
bash-3.1$ chmod ug=rwx,o+x test.txt
bash-3.1$ ls -l test.txt
-rwxrwxr-x 1 cspgcc1 cspg 0 Jan 30 19:40 test.txt
```

- chmod o= test.txt ➔ αφαιρει τα δικαιώματα από Ο group

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Αλλαγή Δικαιωμάτων: Οκταδική μορφή



Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Αλλαγή Ιδιοκτησίας
  - Εντολή *chown* <*new_owner*> <*filename*>
    - Αλλαγή ιδιοκτητησίας ενός αρχείου → **μόνον ο ιδιοκτήτης (και ο su) μπορεί να το πράξει!**
      - Με την αλλαγή, ο νέος ιδιοκτήτης είναι ο μόνος που μπορεί να δώσει τα δικαιώματα πίσω (και ο su).

chown root /directory

Change the owner of /directory to "root". The group of /directory is changed to root's default group (i.e., root).

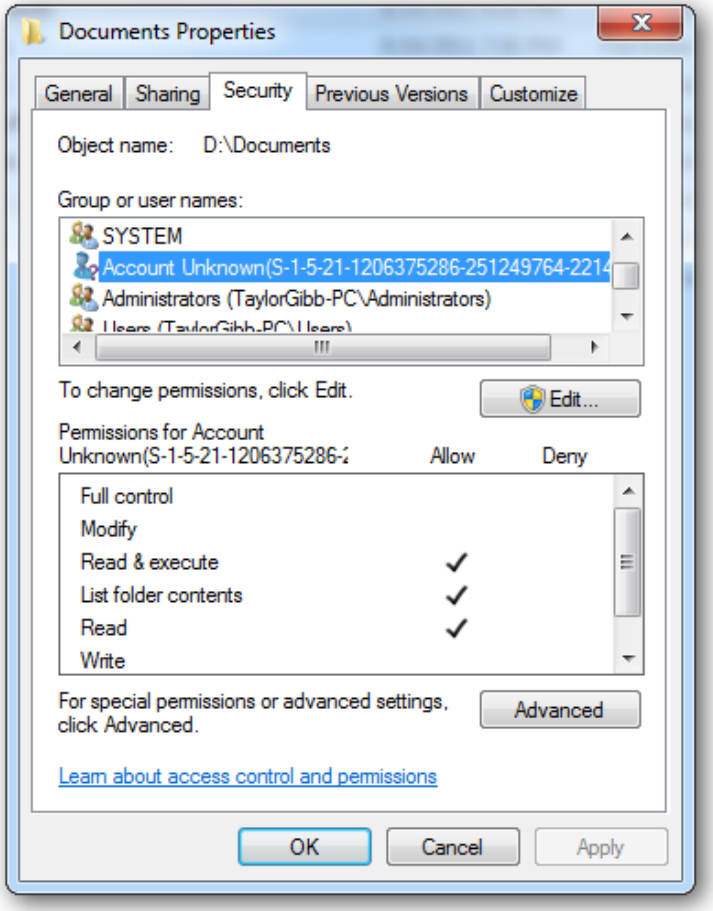
chown root:staff /directory

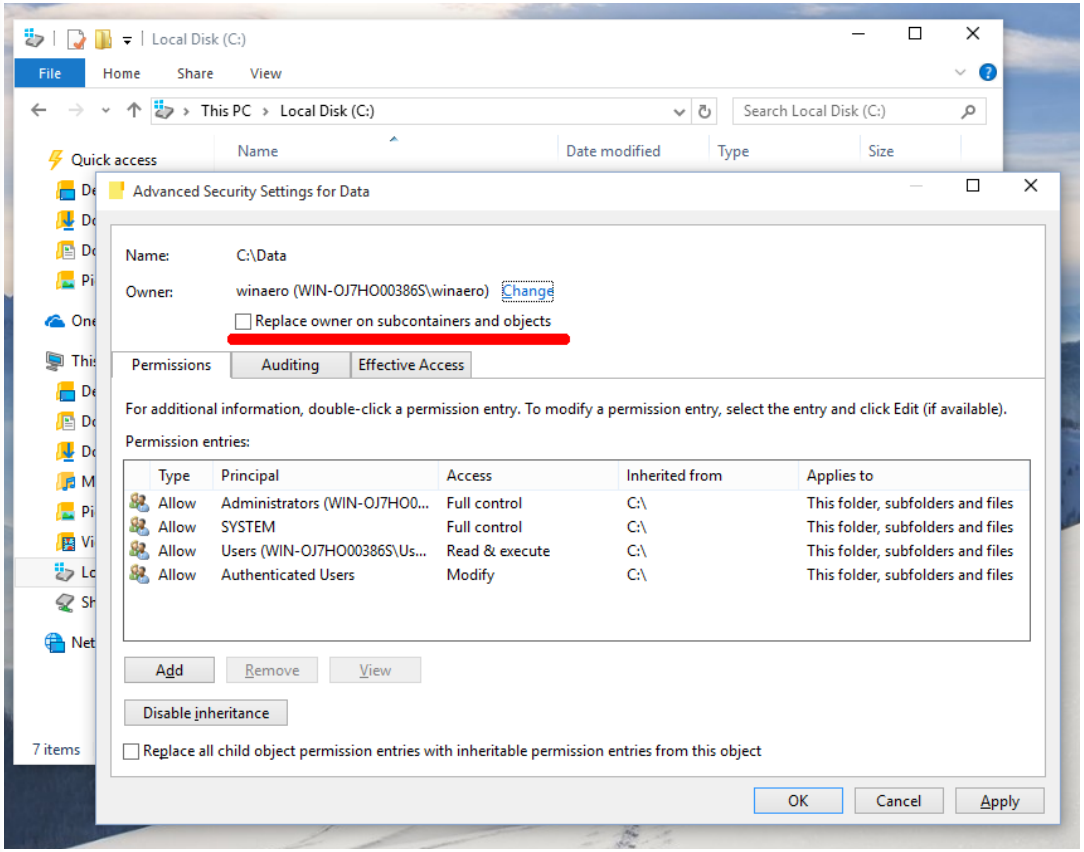
Likewise, but explicity change the group of /directory to "staff".(recall that a user might belong to several groups)

chown -hR root /directory

Change the owner of /directory **recursively (-R)** to "root" (including the **traversal of symbolic links. Used (-h) to exclude them.**)

Taking Ownership in Windows (Fixing Orphaned Users)





Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Αλλαγή Ομάδας
  - Εντολή *chgrp <new_group> <filename>*
    - Αλλαγή ομάδας (effective group) ενός αρχείου σε μια άλλη ομάδα στην οποία ανήκει ο χρήστης
    - Μπορεί να επιτευχθεί και με την chown

```
bash-3.1$ ls -l test.c
-rw-r--r-- 1 cchrys tspecial 55 Mar 17  2015 test.c
bash-3.1$ groups cchrys
cchrys : tspecial epl001 csphd visiting epl371
bash-3.1$ chgrp visiting test.c
bash-3.1$ ls -l test.c
-rw-r--r-- 1 cchrys visiting 55 Mar 17  2015 test.c
bash-3.1$ groups cchrys
cchrys : tspecial epl001 csphd visiting epl371
```

Δείτε Επόμενη Διαφάνεια για βασική & συμπληρωματική ομάδα!

Primary / Supplementary Groups (groups, id)

- Ανά πάσα στιγμή, ένας UNIX χρήστης φέρει ένα user id (uid) και ένα group id (gid).
- Παράλληλα, ένας χρήστης μπορεί να ανήκει και σε άλλα groups, το πρώτο (default) εκ των οποίων ονομάζεται **βασική ομάδα (Primary Group),** ενώ τα υπόλοιπα **συμπληρωματικές ομάδες (Supplementary Groups)**

```
$ groups # print the groups a user is in
faculty epl371 epl132 anyplacegrp epl646 colloqgrp crowdgrp smartpgrp
    smartgrp smartlgrp smartnet tvmgrp
$ id # print group names and their group IDs
uid=1240(dzeina) gid=231(faculty)
    groups=231(faculty),306(epl371),314(epl132),348(anyplacegrp),411(epl646)
    ,426(colloqgrp),446(crowdgrp),453(smartpgrp),466(smartgrp),483(smartlgrp
    ),488(smartnet),505(tvmgrp)
$ newgrp epl371 # αλλαγή βασικής ομάδας
uid=1240(dzeina) gid=306(epl371)
    groups=231(faculty),306(epl371),314(epl132),348(anyplacegrp),41
    1(epl646),426(colloqgrp),446(crowdgrp),453(smartpgrp),466(smart
    grp),483(smartlgrp),488(smartnet),505(tvmgrp)
```

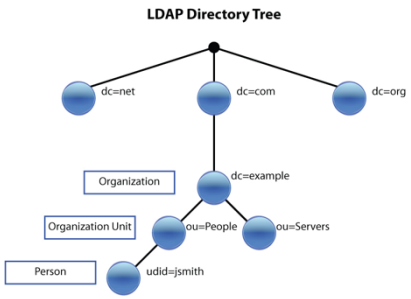
Άντληση Ταυτοτήτων από Εξυπηρετητή Ταυτοποίησης LDAP - Εντολή getent

Παρουσίαση χρηστών (αντίστοιχο του /etc/passwd)

```
$getent passwd
aandre28:*:2923:472:Andreou
Andre:/home/students/cs/2011/aandre28:/bin/bash
acosta01:*:2776:472:Andri
Costa:/home/students/cs/2011/acosta01:/bin/bash
ageorg35:*:2743:472:Anna
Georgiou:/home/students/cs/2011/ageorg35:/bin/bash
```

Παρουσίαση στοιχείων ομάδας (αντίστοιχο του /etc/groups)

```
$getent group cs11
cs11:*:472:
```



Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Δικαιώματα Πρόσβασης αρχείων και καταλόγων κατά τη δημιουργία τους
  - Εντολή *umask (Ορίζει τι δικαιώματα αφαιρούνται)*
  - Τα εξ' ορισμού (προεπιλεγμένα) δικαιώματα ενός δημιουργηθέντος αρχείου ή καταλόγου ρυθμίζονται αρχικά χρησιμοποιώντας μια μεταβλητή **3-ψηφίων** σε οκταδικό σύστημα, που ονομάζεται ***user mask***.
  - Αυτό το ***user mask*** έχει ορισθεί από τον *system* ***administrator*** όταν **δημιουργήθηκε ο λογαριασμός** του κάθε χρήστη.
  - Το *user mask* περιέχει τις **ρυθμίσεις σε οκταδικό** για τα δικαιώματα πρόσβασης που ***ΑΦΑΙΡΟΥΝΤΑΙ*** από το μέγιστο όταν ένας **κατάλογος** ή **αρχείο** δημιουργείται.

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Μέγιστα Δικαιώματα **κατά την δημιουργία**:
  - **Καταλόγου**: *777* (δηλ., rwxrwxrwx)
  - ***Αρχείου***: 666 (δηλ., rw-rw-rw-)
    - Π.χ., με “touch a” τα δικαιώματα του a είναι “666 – umask”
- Γιατί 666 σε αρχεία και 777 σε καταλόγους;
  - Το UNIX προσπαθεί να συμπεριφέρεται έξυπνα όσον αφορά τα ***execute* δικαίωμα**.
    - Πρακτικά τα αρχεία θεωρούνται ότι δεν έχουν ΠΟΤΕ execute δικαιώματα κατά τη δημιουργία τους.
    - Αφαιρώντας εξ'ορισμού το execute σε αρχεία επιτρέπει το σύστημα να έχει περισσότερη ασφάλεια.
    - Το execute σε καταλόγους δεν αφαιρείται εφόσον μπορεί να θέλουμε να παρέχουμε access by file name

Ιδιοκτησία και Δικαιώματα Πρόσβασης

– Παράδεγμα

Το επιπλέον 0 θα εξηγηθεί σε λίγο

```
bash-3.1$ umask
0022
bash-3.1$ touch test.txt
bash-3.1$ ls -l test.txt
-rw-r--r-- 1 cchrys tspecial 0 Jan 31 06:27 test.txt
bash-3.1$ mkdir test-perm
bash-3.1$ ls -ld test-perm/
drwxr-xr-x 2 cchrys tspecial 4096 Jan 31 06:28 test-
  perm/
```

File

666-022 = 644

Directory

777-022=755

Ιδιοκτησία και Δικαιώματα Πρόσβασης

– Παράδεγμα (συνέχεια)

```
bash-3.1$ umask 077  (group & others: NO permissions)
bash-3.1$ touch test2.txt
bash-3.1$ ls -l test2.txt
-rw------- 1 cchrys tspecial 0 Jan 31 06:59 test2.txt
bash-3.1$ mkdir test2-perm
bash-3.1$ ls -ld test2-perm (-d:show dir not content)
drwx------ 2 cchrys tspecial 4096 Jan 31 07:00 test2-
  perm
```

File: 666 - 077 = 600

Directory: 777 - 077 = 700

– Το ***umask*** ρυθμίζεται μια φορά και ισχύει μέχρι την αποσύνδεση του **session**.

- Κάθε φορά που συνδεόμαστε (log in) στο σύστημα, το *umask* κρατά την προεπλεγμένη του τιμή (αργότερα θα μιλήσουμε για το προφίλ)

Ειδικά Δικαιώματα Πρόσβασης

- Σε πολλές περιπτώσεις προκύπτει η ανάγκη για προσωρινά αναβαθμισμένα δικαιώματα.
  - Π.χ., ο χρήστης dzeina:faculty να έχει δικαιώματα root:root κατά την εκτέλεση την εντολής /usr/bin/passwd
- Υπάρχουν οι ακόλουθοι τρόποι:
  1. Login as <newuser> (π.χ., su – newuser)
  2. Run command with sudo (π.χ., sudo –u newuser /usr/bin/passwd, requires the existence of configurations in /etc/sudoers/)

Use the following to avoid annoying password prompt – useful in scripts:

$ sudo visudo

[username] ALL=(ALL) NOPASSWD: ALL

1. Special Permissions on Executable (Ειδικά Δικαιώματα)
   - **Set User ID (SUID)** -- για εκτελέσιμα αρχεία
   - **Set Group ID (SGID)** -- για εκτελέσιμα αρχεία
   - **Sticky bit (STB)** -- για καταλόγους

Ειδικά Δικαιώματα Πρόσβασης: **Αρχεία**

Ενδέχεται να είναι και στο τέλος

- Ρύθμιση Ειδικών Δικαιωμάτων
  - Χρήση της εντολής *chmod* σε οκταδική μορφή:
    - π.χ. *chmod **7**777 filename*

suid	**sgid**	stb	r	w	x	r	w	x	r	w	x
4	**2**	**1**	4	2	1	4	2	1	4	2	1
7 **Special**			7 user			7 group			7 others		

Ειδικά Δικαιώματα Πρόσβασης: **Αρχεία**

- Ειδικά Δικαιώματα: **Set User ID (SUID)**
  - ***SUID*** επιτρέπει στους χρήστες να εκτελέσουν ένα αρχείο και να γίνουν **προσωρινοί ιδιοκτήτες** (Owners) του αρχείου (κατά τη διάρκεια της εκτέλεσης).
    - Στο Linux / Unix αγνοείται για καταλόγους (μόνο για αρχεία)
    - **Παράδειγμα:** Η εντολή *passwd* ή ping με ιδιοκτήτη τον *root* έχει τις ακόλουθες ειδικές ρυθμίσεις:

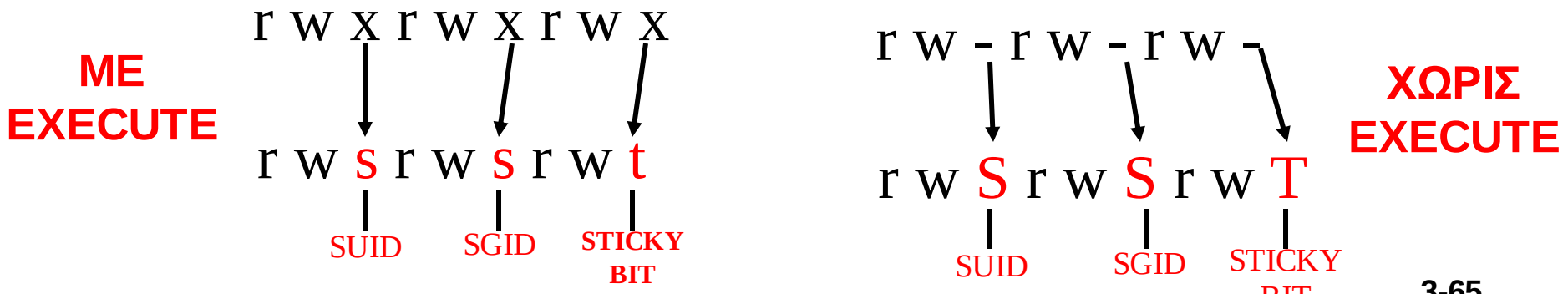
```
bash-3.1$ ls -l /usr/bin/passwd
-r-s--x--x 1 root root 21944 Feb 12  2006 /usr/bin/passwd
bash-3.1$ ls -al /bin/ping
-rwsr-xr-x 1 root root 40760 Sep 26  2013 /bin/ping
```

SUID

    - Όταν ένας χρήστης εκτελεί την εντολή *passwd*, ο χρήστης γίνεται προσωρινά ο «*root*» χρήστης για **όσο τρέχει η εντολή** (δηλ., η διεργασία θα έχει τα **ίδια δικαιώματα** όπως αυτά του **ιδιοκτήτη** του αρχείου)
    - Θα δούμε σε λίγο πως τίθεται η επιλογή «s»

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- **Παρουσίαση Ειδικών Δικαιωμάτων**
  - Η κατάσταση των δικαιωμάτων πρόσβασης που εμφανίζεται με την εντολή *«ls -l»* **δεν έχει ξεχωριστό τμήμα για τα ειδικά δικαιώματα σε πολλές υλοποιήσεις ☹.**
  - Επειδή τα ειδικά δικαιώματα απαιτούν συνήθως ***«execute»,*** καλύπτουν/αντικαθιστούν το **δικαίωμα *execute*** στην παρουσίαση της εντολής *«ls -l»*.



Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Ειδικά Δικαιώματα: **Set Group ID (SGID)**
  - Η ***SGID*** κάνει τους χρήστες να **γίνουν μέλος της ομάδας** του γονικού καταλόγου
  - Χρήσιμο για δημιουργία κοινόχρηστης πρόσβασης σε άτομα από διαφορετικές ομάδες (π.χ.,student, faculty)
    - Υπάρχει κάποια ομάδα "GALL" με μέλη τους A:G1, B:G2, C:G3 (username:group)
    - Ο κατάλογος /projects/ ανήκει στο GALL.

suid	sgid	stb
4	2	1
7		
Special		

Ο A:G1 εκτελεί «touch file» (δημιουργία αρχείου)

- Τώρα το **file** είναι του **A:G1** (δείτε ls –al file)

Ενώ εάν εκτελούσαμε το πιο κάτω πριν το touch:

- chmod **2**777 /projects/
- Τότε το **file** θα είχε δικαιώματα **A:GALL**

Ιδιοκτησία και Δικαιώματα Πρόσβασης

Παράδειγμα SGID

$ls -ald anyplace/ # d shows directory not content

drwxrw**s**r-x 21 kgeorg10 anyplacegrp 4096 Jan 2 00:02 anyplace/

Πλέον, ότι αρχεία δημιουργούνται ανήκουν στο group anyplacegrp στο οποίο ανήκουν και οι δυο χρήστες (βολικό για αλλαγές και από τους δυο)

$ ls -al

total 180

drwsrwsr-x 21 kgeorg10 anyplacegrp 4096 Jan 2 00:02 .

drwxr-xr-x 68 root   root   8192 Sep 30 10:02 ..

drwsr-sr-x 2 **dzeina   anyplacegrp** 4096 Sep 19 14:57 architect

drwsr-sr-x 2 dzeina   anyplacegrp 4096 Sep 19 14:57 contact

-rwxrw-rw- 1 dzeina   anyplacegrp   181 Sep 18 13:55 contact.html

drwxrwxr-x 2 **kgeorg10 anyplacegrp** 4096 Nov 27 13:14 css

Ιδιοκτησία και Δικαιώματα Πρόσβασης

- Ειδικά Δικαιώματα: **Sticky Bit (STB)**
  - Εάν εφαρμοστεί το ***Sticky Bit*** τότε η **διαγραφή αρχείων/καταλόγων** μπορεί να γίνει από **ένα χρήστη μόνο** σε αρχεία που έχει **προσθέσει ο ίδιος ο χρήστης**.
    - Το *Sticky Bit* εκτελεί μια χρήσιμη λειτουργία στους καταλόγους, π.χ., στο /tmp (κοινόχρηστος χώρος)
  - Παράδειγμα:

```
bash-3.1$ ls -ld /tmp
drwxrwxrwt 71 root root 16384 Jan 31 04:10 /tmp
```

Sticky Bit

  - Εάν το /tmp ήταν απλά 777 τότε θα μπορούσε οποιοσδήποτε να διαγράφει καταλόγους/αρχεία άλλων χρηστών (όχι μόνο προσωπικούς).

Προσθήκη Τοπικών Χρηστών (Εντολή useradd – LDAP luseradd / lgroupadd)

- **# less /etc/default/useradd**
  GROUP=100
  HOME=/home
  INACTIVE=-1
  EXPIRE=
  SHELL=/bin/bash
  SKEL=/etc/skel
  CREATE_MAIL_SPOOL=yes

To change the **default home directory** location for all new users

```
# useradd -D -b /opt/users
```

To change the **default login shell**

useradd -D -s /bin/sh

Create multiple users with same UID

```
# useradd -o chrisk -u 501
# useradd -o chris -u 501
# useradd -o user -u 501
```

Verify the UID of the newly create users

```
# grep 501 /etc/passwd
chrisk:x:501:501::/home/chrisk:/bin/sh
chris:x:501:504::/home/chris:/bin/sh
user:x:501:505::/home/user:/bin/sh
```

Manually assign a UID to the user

useradd -u 550 chrisk

Add user to different primary group

useradd -g admin,dba chris