Users and Groups

- if a user executes programs, they use the configs in his home direcory: `~/.pogram`, `~/.config/program` etc.
- if a superuser executes programs, they use the configs in `/usr/share/program` etc.
- all files and directories in the home directory (`~`) should belong to the user of the home directory and the user's group (`sudo chown areo:users ./Documents -R`)
- all files and directories in the root directory (`/`) besides the home directory should belong to the superuser `root` and it's group `root`

Important files

- `/etc/shadow`: Secure user account information
- `/etc/passwd`: User account information
- `/etc/gshadow`: Contains the shadowed information for group accounts
- `/etc/group`: Defines the groups to which users belong
- `/etc/default/useradd`: Default values for account creation
- `/etc/shells`: contains the full pathnames of valid login shells
- `/etc/default/useradd`: useradd defaults

Get overview of Users / Groups

- `who`: lists users currently logged on the system
- `sudo passwd -Sa`: list all existing user accounts including their properties stored in the **user database**
- `find /path -group groupname / groupnumber` and `find /path -user user`: list files owned by a user or group
- `groups user`: display group membership
- `id user`: provides additional detail, such as the user's UID and associated GID
- `cat /etc/group`: list all groups on the system

Add Users / Groups

- `sudo useradd -m (-g username) -G additional_groups -s login_shell username`
    - `-m/--create-home`: creates the user home directory as /home/username. Within their home directory, a non-root user can write files, delete them, install programs etc.

    - `-g/--gid`: the group name or number of the user's **initial login group** (primary group)
        - the group name or number must exist
            - the default behaviour is to create a group with the same name as the username (if `USERGROUPS_ENAB` variable is set to `yes` in `/etc/login.defs`, else `useradd` will set the primary group of the new user to the value specified by the `GROUP` variable in `/etc/default/useradd`)
            - using a single default group (e.g. users) for every user is not recommended. The reason is that typically, the method for facilitating shared write access for specific groups of users is setting user umask value to `002`, which means that the default group will by default always have write access to any file you create. In the recommended scenario, where the default group has the same name as the user name, all files are by default writeable only for the user who created them. Setting the setgid bit on a directory to allow write access in a specific folder
    - `-G/--groups`: list of **supplementary groups** (secondary groups) which the user is also a member of. Each group is separated from the next by a comma, with no intervening spaces: `GROUP1,GROUP2,...`.
        - it must refer to an already existing group
        - the default is for the user to belong only to the initial group
    - `-s/--shell`: defines the path and file name of the user's default login shell. After the boot process is complete, the default login shell is the one specified here. Sets the SHELL variable in `/etc/default/useradd`
        - the default value used for the login shell of the new account can be displayed using `useradd --defaults`. The default is Bash
        - in order to be able to log in, the login shell must be one of those listed in `/etc/shells`
        - do not use the `/usr/bin/bash` path instead of `/bin/bash`, unless it is properly configured in `/etc/shells`
- `passwd` or `passwd archie` (from another user): protect the newly created user archie with a password
- `groupadd group`: create new groups
- `gpasswd -a user group`: add users to a group
    - `usermod -aG additional_groups username`: adds a user to additional groups (replace `additional_groups` with a comma-separated list. If the `-a` option is omitted in the usermod command above, the user is removed from all groups not listed in `additional_groups`

Delete Users / Groups

- `userdel -r username`: delete user accounts
  - ithe `-r` option specifies that the user's home directory and mail spool should also be deleted
- `groupdel group`: delete existing groups
- `gpasswd -d user group`: remove users from a group

Change Users / Groups

- `sudo chown areo:users ./Documents -R`: (**ch**ange **own**er) change file's owning user and group

  - `-R` (`--recursive`): auch Unterverzeichnisse und Dateien darin

- `newgrp groupname`: If a GID change is required temporarily one can also use the `newgrp` command to change the user's default GID to another GID at runtime

  - to change back to the default GID, execute `newgrp` without a groupname.

- `usermod -d /my/new/home -m username`: change a user's home directory

  - the `-m` option also automatically creates the new directory and moves the content there
  - <u>other way</u>: one can create a link from the user's former home directory to the new one. Doing this will allow programs to find files that have hardcoded paths: `ln -s /my/new/home/ /my/old/home`

- `usermod -l newname oldname`: change a user's login name

  - one should make certain that one is not logged in as the user whose name one is about to change. One should open a new tty (`ctrl+alt+fX`) and log in as root or as another user and elevate to root

- `usermod -s /bin/bash username` or `chsh -s /bin/bash`: to change the user's login shell

  - `cat /etc/shells` or `chsh -l`: list all installed shells

- `groupmod -n new_group old_group`: rename `old_group` group to `new_group`

  > - if the user is currently logged in, they must log out and in again for the change to take effect

Superuser (root)

- has complete access to the operating system and its configuration; it is intended for administrative use only.
- unprivileged users can use the `su` and `sudo` programs for controlled privilege escalation
    - `wheel`: Administration group, commonly used to give privileges to perform administrative actions. It has full read access to journal files and the right to administer printers in CUPS. Can also be used to give access to the `sudo` and `su` utilities
    - for this the `/etc/sudoers` file has to be edited via the `sudo EDITOR=nvim visudo`
        - uncommenting `%wheel ALL=(ALL) ALL`: members auf group wheel can execute any command with `sudo`
        - uncommenting `%wheel ALL=(ALL) NOPASSWD: ALL`: members auf group wheel can execute any command with `sudo` without typing in a password
        - adding `%wheel ALL=(ALL) NOPASSWD: /usr/bin/mount,/usr/bin/umount,/usr/bin/pacman -Syu,/usr/bin/systemctl restart NetworkManager`: members auf group wheel can execute the specified commands with `sudo` without typing in a password

          > - one has to login and out in between to make it working
- `passwd:` changes root password, when executed as root

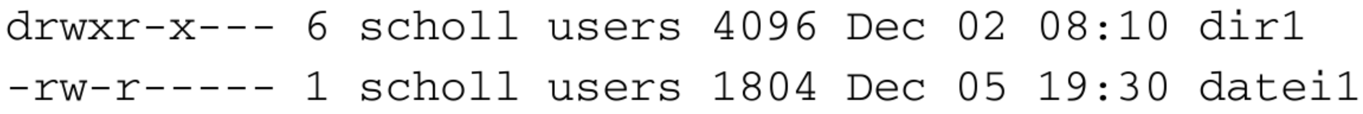
Permissions

- home directory: at most 755 (`drwxr-xr-x`)
    - should not be writeable by the group or others

Rechte:

- `r` steht für Leserechte bei Datei / Verzeichnis (man darf ins Verzeichnis reinschauen)
- `w` steht für Schreiberechte bei Datei / Verzeichnis (Dateien anlegen, umbenennen, löschen)
    - der Name der Datei steckt in der Informationsstruktur seines Verzeichnis und ist nicht Inhalt der Datei
- `x` steht für Ausführungsrechte bei Datei / Verzeichnis (man darf reinwechseln und Objekte benutzen)

Aufbau: `ls -l` (als Liste), `ls -al` (versteckte Dateien) oder `ls -alh` (human readable)



```
drwxr-x--- 6 scholl users 4096 Dec 02 08:10 dir1
-rw-r----- 1 scholl users 1804 Dec 05 19:30 datei1
```

1. Typ des Eintrags: `d` = directory, `-` = file oder `l` = link
2. Rechte für Benutzer, Gruppe und Andere als Tripel mit entweder Recht vorhanden: `w` oder nicht: `-`
3. Anzahl Einträge bei Verzeichnis / Anzahl Hardlinks bei Datei
4. ...

Zugriffsrechte ändern (`chmod` , change mode):

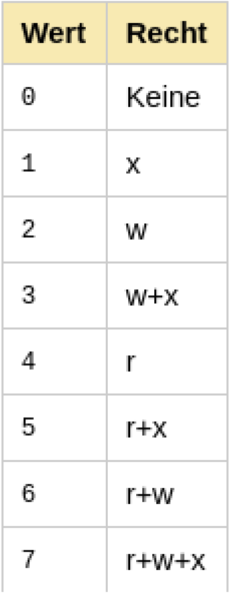
- im **Symbolischen Modus** codiert: `chmod go-rw file1.txt file2.txt`

    1. Benutzertypen, deren Rechte verändert werden sollen: `u` (user), `g` (group), `o` (others) oder `a` (all)
    2. `+` Rechte setzen, `-` Rechte entziehen, `=` nur die explizit angegebenen Rechte setzen und die restlichen entziehen
    3. Menge an Rechten `r` , `w` oder `x`

- im **Oktal-Modus** codiert: `

    - `chmod 777 deCapitalizer.sh pdf_to_pic_v2.sh reverser_v2.sh`

    - `chmod 740 -R myBachelorThesis`



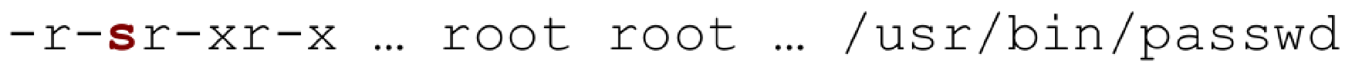
Wert	Recht
0	Keine
1	x
2	w
3	w+x
4	r
5	r+x
6	r+w
7	r+w+x

- `-R` (`--recursive`): auch Unterverzeichnisse und Dateien darin

Sonderrechte

SUID - set user ID

`ls -alh`:

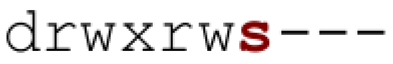


```
-r-sr-xr-x … root root … /usr/bin/passwd
```

- Setzen des Bits: `chmod u+s file.txt`
- Ausführung eines SUID-Prozesses unter der UID und damit mit den Rechten des Besitzers der Programmdatei (anstatt mit Rechten des ausführenden Benutzers)
- Beispiel oben: Beim Aufruf von `passwd` läuft das Programm mit der uid des Besitzers root und darf in `/etc/shadow` schreiben

SGID - set group ID

`ls -alh`

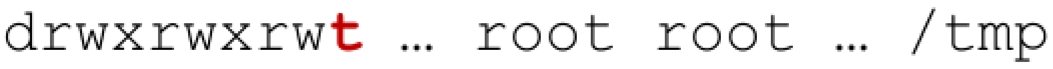


```
drwxrws---
```

- Setzen des Bits: `chmod g+s verzeichnis`
- Dateien: Ausführung eines SGID-Prozesses mit der GID und damit mit den Rechten der Gruppe, der die Programmdatei gehört (anstatt mit den Rechten der Gruppe, die ausführt)
- Verzeichnisse: Neu angelegte Dateien gehören der Gruppe, der auch das Verzeichnis gehört (anstatt der Gruppe, die eine Datei erzeugt)

SVTX - save text bit or sticky bit

`ls -alh`

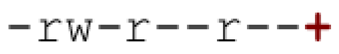


```
drwxrwxrwt … root root … /tmp
```

- zum Löschen der Datei in einem Verzeichnis muss der Benutzer auch der Eigentümer der Datei oder des Verzeichnisses sein

Access Control Lists (ACL's)

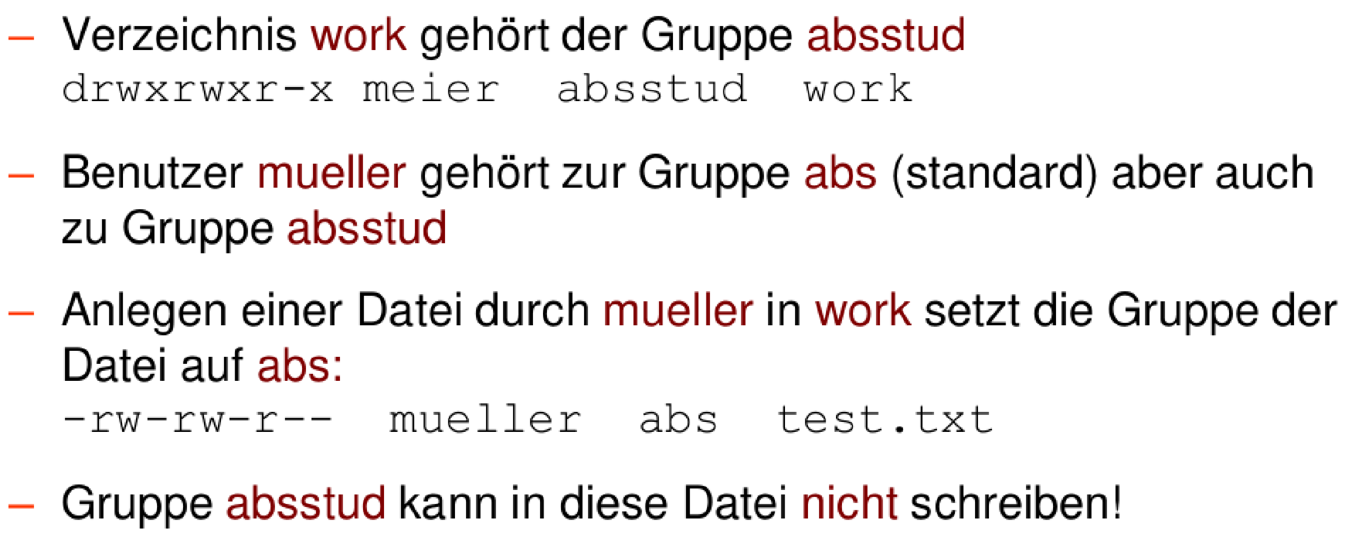
`ls -alh`



```
-rw-r--r--+
```

- einzelnen Nutzerinnen (oder auch Gruppen) können gezielt Rechte an einzelnen Dateien gegeben bzw. entzogen werden

Anwendungsbeispiel für SGID



– Verzeichnis work gehört der Gruppe absstud

```
drwxrwxr-x meier  absstud  work
```

– Benutzer mueller gehört zur Gruppe abs (standard) aber auch zu Gruppe absstud

– Anlegen einer Datei durch mueller in work setzt die Gruppe der Datei auf abs:

```
-rw-rw-r--  mueller  abs  test.txt
```

– Gruppe absstud kann in diese Datei nicht schreiben!

1. mit `newgrp` die aktuel aktive Gruppe in `absstud` ändern
2. `chmod g+s work`
3. Mueller setzt Schreibrechte für alle Benutzer