TM890

LINUX basics

PERFECTION IN AUTOMATION
www.br-automation.com

Requirements	
Training modules	None
Software	Linux
Hardware	1 Control Computer

Table of contents

Table of contents

1 Introduction

APROL uses the SUSE Linux operating system (SLES), because its performance and stability make it destined to be deployed as a process control system.

The following general conditions must be fulfilled:

- Operators need no operating system knowledge.
- Standard engineering does not necessarily require operating system knowledge, but a basic knowledge is surely helpful.
- System administrators require detailed Linux knowledge

Basic knowledge, tips, and tricks are provided in this documentation. This documentation is also partly valid for other Linux distributions.

Figure 1: SUSE Linux Enterprise Server (SLES) - Let's go

1.1 Training module objectives

This 'Linux basics' should help you to manage the powerful Linux operating system. All of the important structures and Linux commands are mentioned in this documentation, which serves well as a source for reference.

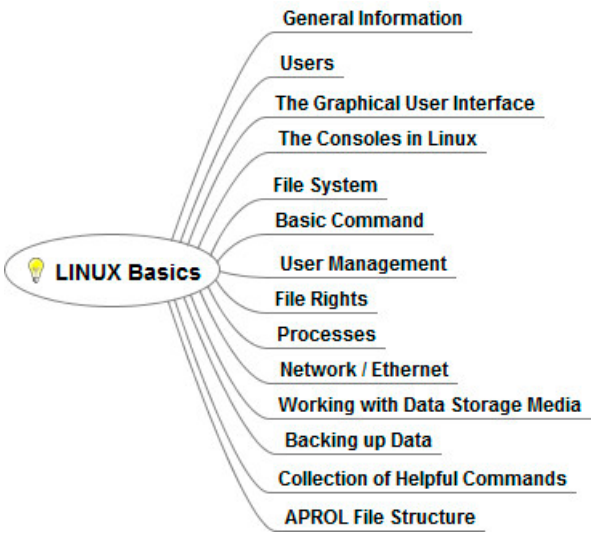


Figure 2: Overview

2 General Information

Linux is an operating system similar to Unix and is actually only composed of the operating system kernel. The complete source code is freely available and can be changed, whereby it is a completely new software development method (open source).

The efficient, flexible, multi-user operation, and availability of all network protocols is a great feature in comparison with the Windows operating system. This allows several users to operate simultaneously on one computer with a multi-user system. This functionality is normally accessed over a network connection. Linux manages the different process and allows the users to work independently from each other.

Cluster solutions in a Linux working environment are the favored architecture in the IT word, so that demands such as high availability, load balancing, and grid computing are covered by the main competences of the operating system.

Bonding allows each computer with two or more network cards to double the communication routes. Physical network cards are depicted as one virtual network card.

Apache web server with PHP("PHP: Hypertext Preprocessor"), and modular structure (phases/handlers) has been seen as one of the most powerful web servers on the market for years. Flash is integrated as a development environment to create multi-media content, so-called Flash films in 'Small Web Format' and(or 'Shock Wave Flash', and also support interactive content.

Apart from parallel compiling on a local computer, the GCC (GNU Compiler Collection) also allows a distribution of the compile jobs through the 'Icecream' technology to other computers in the network (increase in the compiling speed).

Virtual Network Computing Software composed of VNC server and VNC viewer runs on different operating system platforms, e.g. A Linux desktop can be accessed from a Windows desktop. VNC also exists as Java variant and opens a 'Remote control' from a web browser without having to install a program on the machine. It should be noted that several 'Remote control' sessions can be started in Linux without having an influence on each other (Linux has a real multi-user functionality).

3 Users

Users must log in before they can use the system. A user name and password are required for this. The login screen after booting can be seen in the following.

Users can work according to their rights after having logged in.

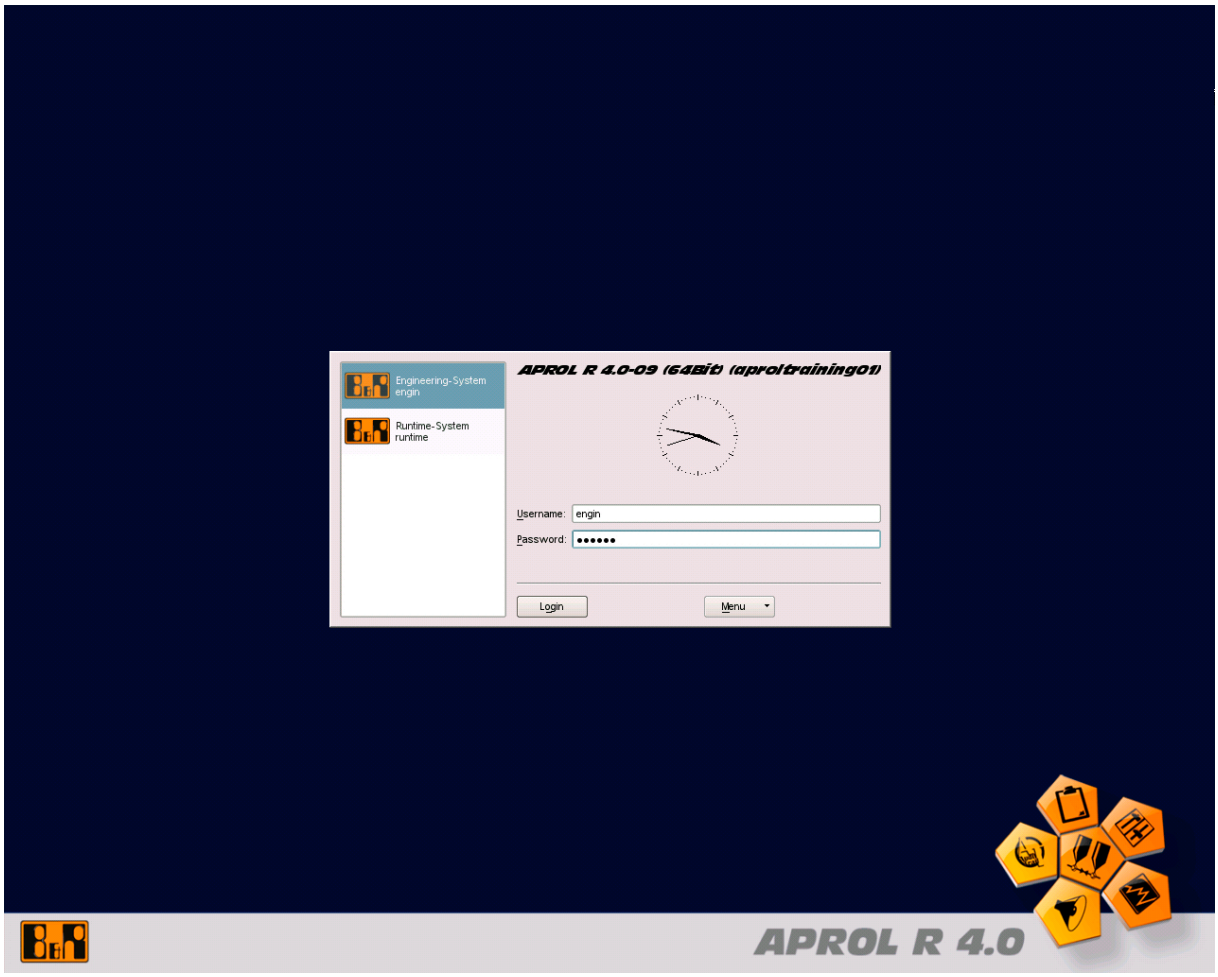


Figure 3: User login in a graphical console

4 The Graphical User Interface

In the past, the operating system was started and the graphical user interface was then loaded from the prompt. This differentiation is no longer made for most operating systems. The graphic interface has now more or less become a part of the operating system.

Linux behaves in another way. There is a clear differentiation between kernel and graphical user interface. It is possible to decide in which 'operating mode' the system should start by selecting the individual runlevels (0-5). This means that a 'server operation' without graphic is possible.

4.1 The Linux Runlevels

Linux computers are started in a structured manner. This means that one runlevel is started after the other. You specify which runlevel should be booted to in YaST (Yet another Setup Tool).

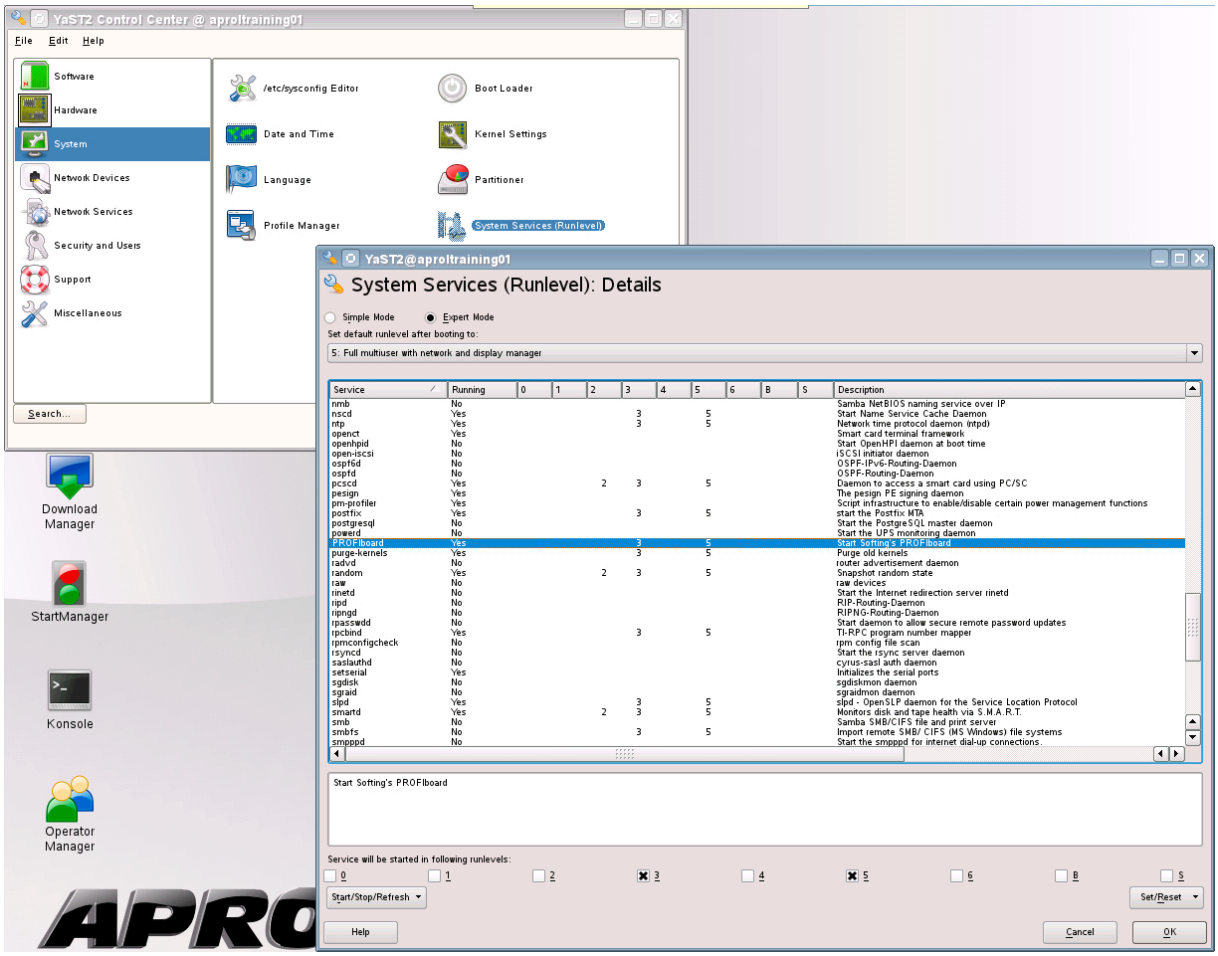


Figure 4: Configuration of runlevel 2 with YaST2

The following runlevels are possible:

RUNLEVEL	Description
0	Shutdown (computer is shut down completely)
1	Single user operation only with local resources (no network)
2	Local multi-user operation without complete network support
3	Multi-user operation with complete network support. There is no graphical user interface.
4	Currently not used at present
5	Multi-user operation with network; with an additional graphical user interface.
6	System reboot

The graphic shown above indicates which program is started or stopped in which runlevel. This provides the perfect mechanism for starting applications in the correct order.

4.2 The Graphics in Linux

As mentioned, there is a clear differentiation between kernel and graphical user interface in Linux. This user interface (mainly an X window system) is therefore not a fixed part of the operating system and must/can also be started such as all other programs.

A so-called window manager is built up on the X window system. The window manager is the basis for a desktop environment such as KDE (K Desktop Environment) and its window-oriented user guidance.

This environment offers many different programs to work more comfortably. Everything which is available in MS-Windows, such as calculation programs, file explorer, text processing, etc. is available and also APROL.

These programs work as expected.

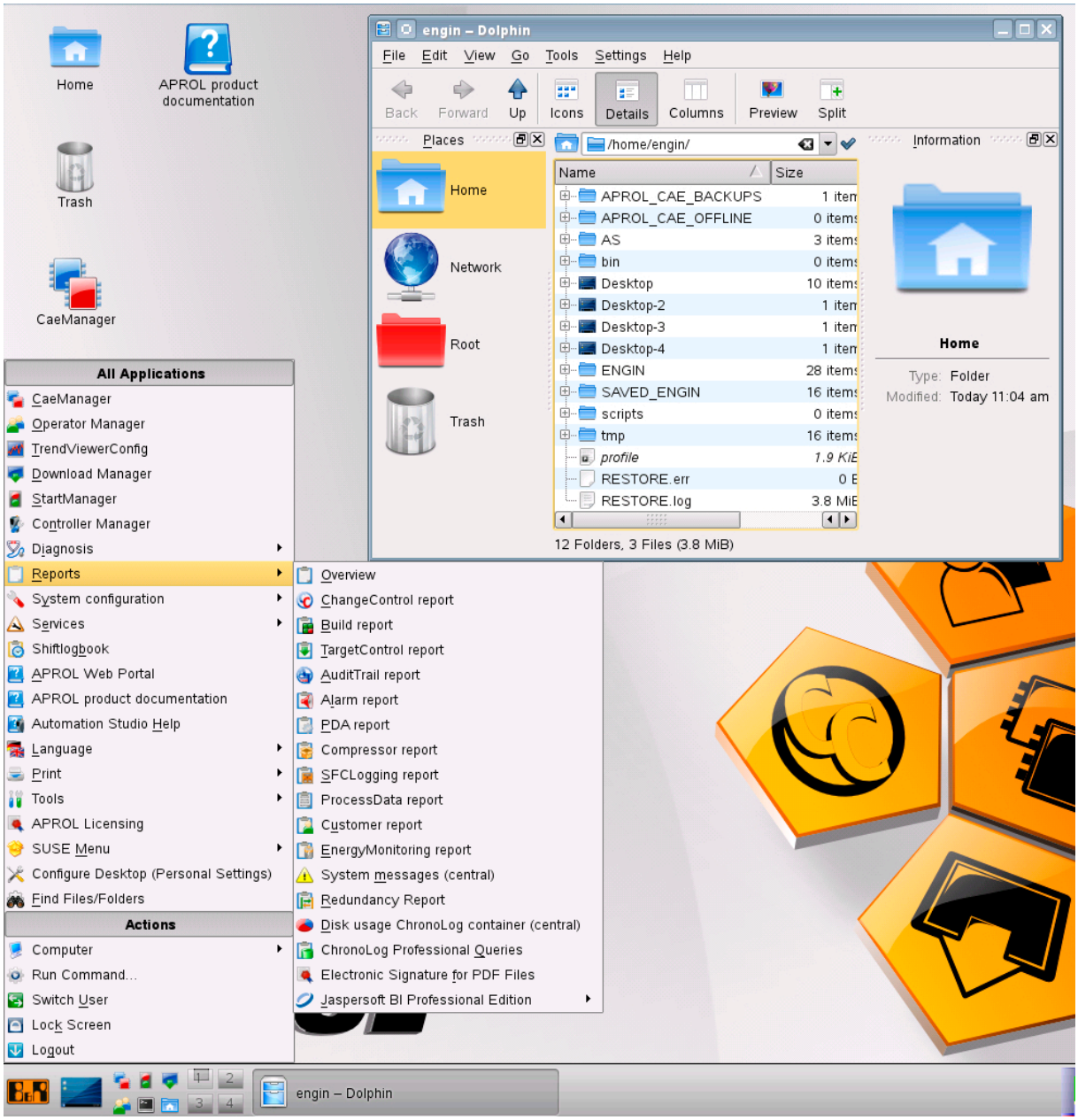


Figure 5: KDE look and feel

5 The Consoles in Linux

The system starts to runlevel 5 after a Linux installation if nothing else has been configured. You then have 6 text consoles and one graphic console available.

Text consoles:

The text consoles can be reached with the F1 to F6 function keys. The respective key combinations are 'Ctrl Alt F1 (to F6)'. You can log in to these consoles and execute commands (similar to the command prompt). Another graphical user interface can be started on the computer, which is entered as name or IP address, with the following command.

```
X :1 -ac -query <name or IP address> -dpi 75
```

The value in the command (X :**1**) is 'variable', i.e. can also be 2, 3, etc. in order to be able to start further graphic consoles. Further graphic user interfaces are started on the consoles F8, F9, F11, and F12. This means that there are five graphic consoles available on each computer.

Console F10 is reserved for Linux. Kernel messages which may be of interest appear here.

Graphical console:

The graphical console is normally found on F7. As described above, other graphic consoles can be started using the corresponding command. It is not necessary to use the command mentioned above to start a new graphic console. This can also be done directly in KDE with **<change user>/<start new session>**. The key combination **<Ctrl>+<Alt>+<F number console>** or the above mentioned menus change between consoles.

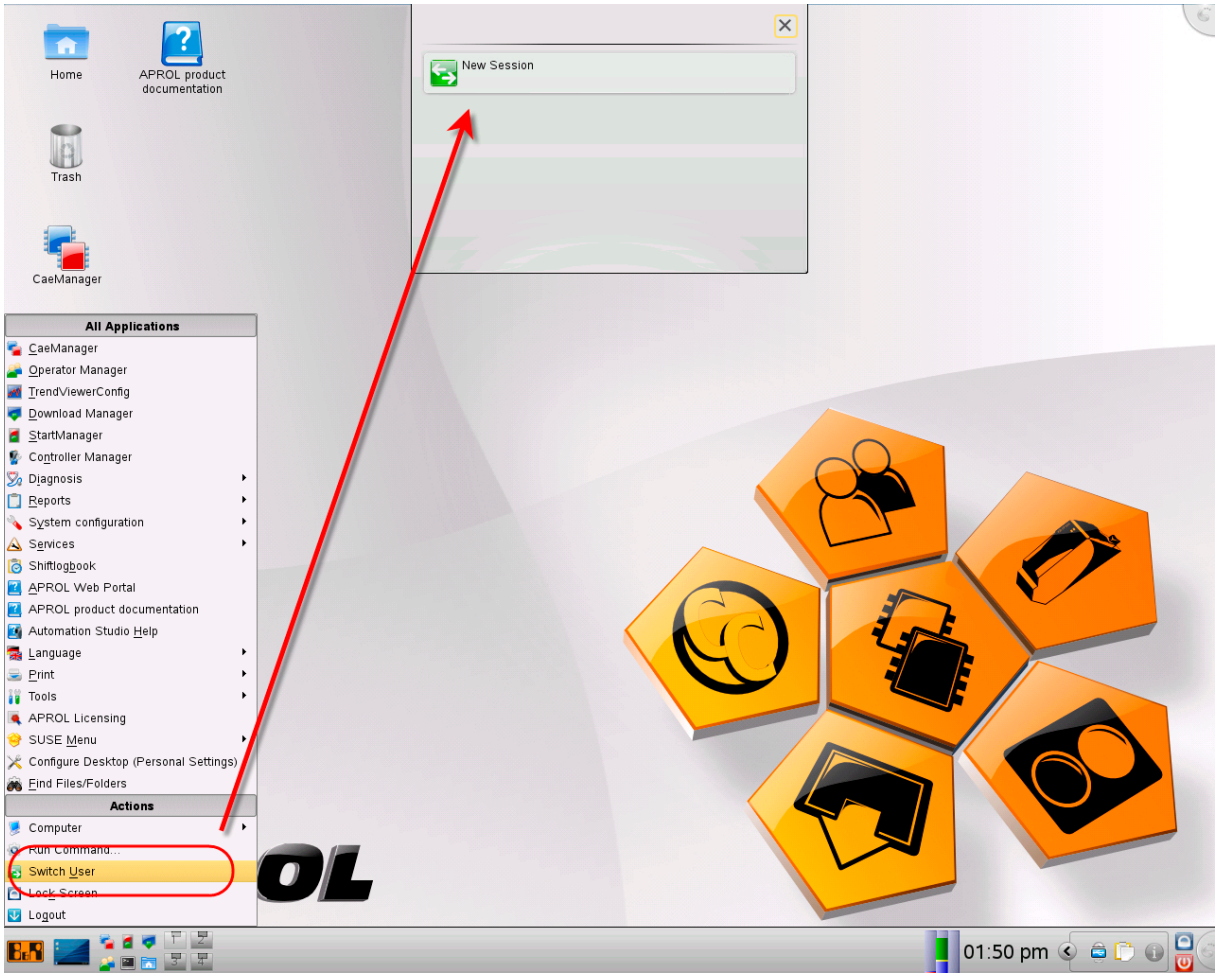


Figure 6: Start another local graphical user interface

The operation can be reviewed and confirmed in the following dialog.

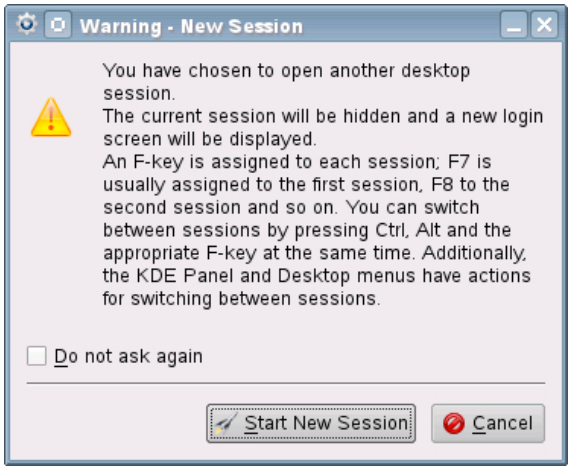


Figure 7: Security query and information when starting the new graphic

6 File System

There are many different file systems in Linux. For example, reiserfs, EXT2, EXT3, XFS, …

The journaling file systems are interesting and offer great error tolerance. This is positive if there is a hard computer power-off, because no exhaustive checks must be carried out. XFS and EXT3 are such file systems.

Long file names can always be processed in the Linux file system. The users can name their files as they like. There is no need to use certain suffixes (.txt, .doc, ...), because the system recognizes the type with internal markings. Normal ending can be used for a better overview.

7 Directory Structure

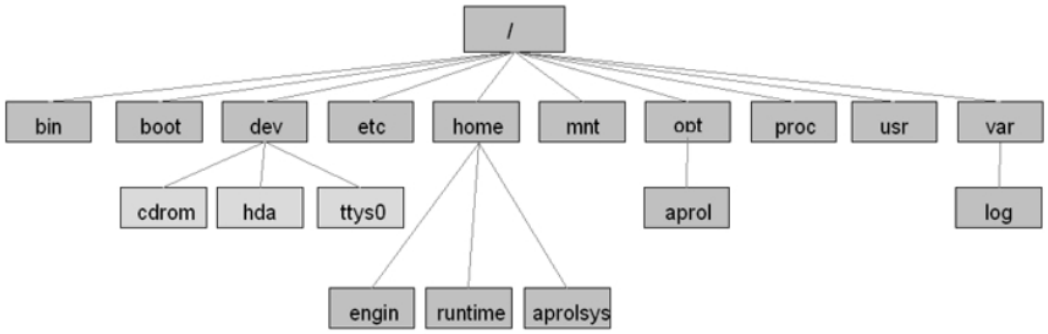


Figure 8: Directory structure overview

The graphic shows a part of the Linux directory structure in APROL. The uppermost entry "/" is the root directory.

The users directories are in the "/home/" directory. This represents the home directory of the user. The user starts in their home directory after having logged in.

The user can carry out various required actions. Settings which are then changed only affect that user. (desktop, background, font color, ….) there are no changes at operating system level. Changes within the operating system can only be carried out by the superuser "root" if this is necessary. Each user's home directory is a private area which can be set up as desired.

7.1 Description of the Directories

The following table describes the Linux standard directories. They can be used by various users, depending on the rights, and new directories can be created. Normally, these directories remain untouched. Own structures are normally only created by the responsible user.

Directory	Description
/bin	This directory contains elementary Linux commands in the form of executable files. All of the command here can be executed by all users, and not only by the user "root".
/sbin	The directory contains Linux commands which are only allowed to be executed by the user "root".
/boot	This contains the files required by the Bootloader, LILO, and GRUB. Other kernel versions can be stored here.
/dev	Contains all device files
/etc	Almost all of the system's configuration files can be found here. For example, the files here control the boot process or configure the network.

Directory	Description
/home	The user-specific directories and the personal data are saved here. If a new user is created, a directory corresponding to the user name will be created in the /home directory. The directories are created with the "AprolConfig -createsystem" script in APROL.
/lib	Library directory The system drivers are in /lib/modules.
/lib64	Library directory (64 bit)
/mnt	External file systems can be mounted in this directory. E.g. CD-ROM drive
/opt	APROL is installed in this directory. Furthermore, KDE is installed here.
/proc	The directory is loaded virtually as a runtime directory by the system. It contains a complex image of the system with all current processes. The information can be used for diagnostic purposes.
/sys	Configuration settings created by the system are stored in the directory.
/usr	The abbreviation stands for "Unix System Resources" (previously "user"). Most of the system tools, libraries, etc. are saved here. Only non-changeable files (static) should normally be stored here. The non-static files are stored in /var.
/var	A large number of subdirectories with changeable files are stored here. These subdirectories often have the same name as those in /usr, whereby the files in the /var subdirectories are allowed to be changed (dynamic).
/lost&found	Data which is buffered during an eventual system crash is stored here.
/media	This is the media directory. All external media which can be mounted by the system can be viewed here. E.g. CD-ROM, floppy, usbdisk
/root	All configuration data and application data for the superuser "root"
/srv	Data for system services
/tmp	Directory for temporary files

7.2 Navigation in the Directory Structure

Navigation in the graphical user interface is comfortable with Dolphin. This is the equivalent to the well-known Windows explorer.

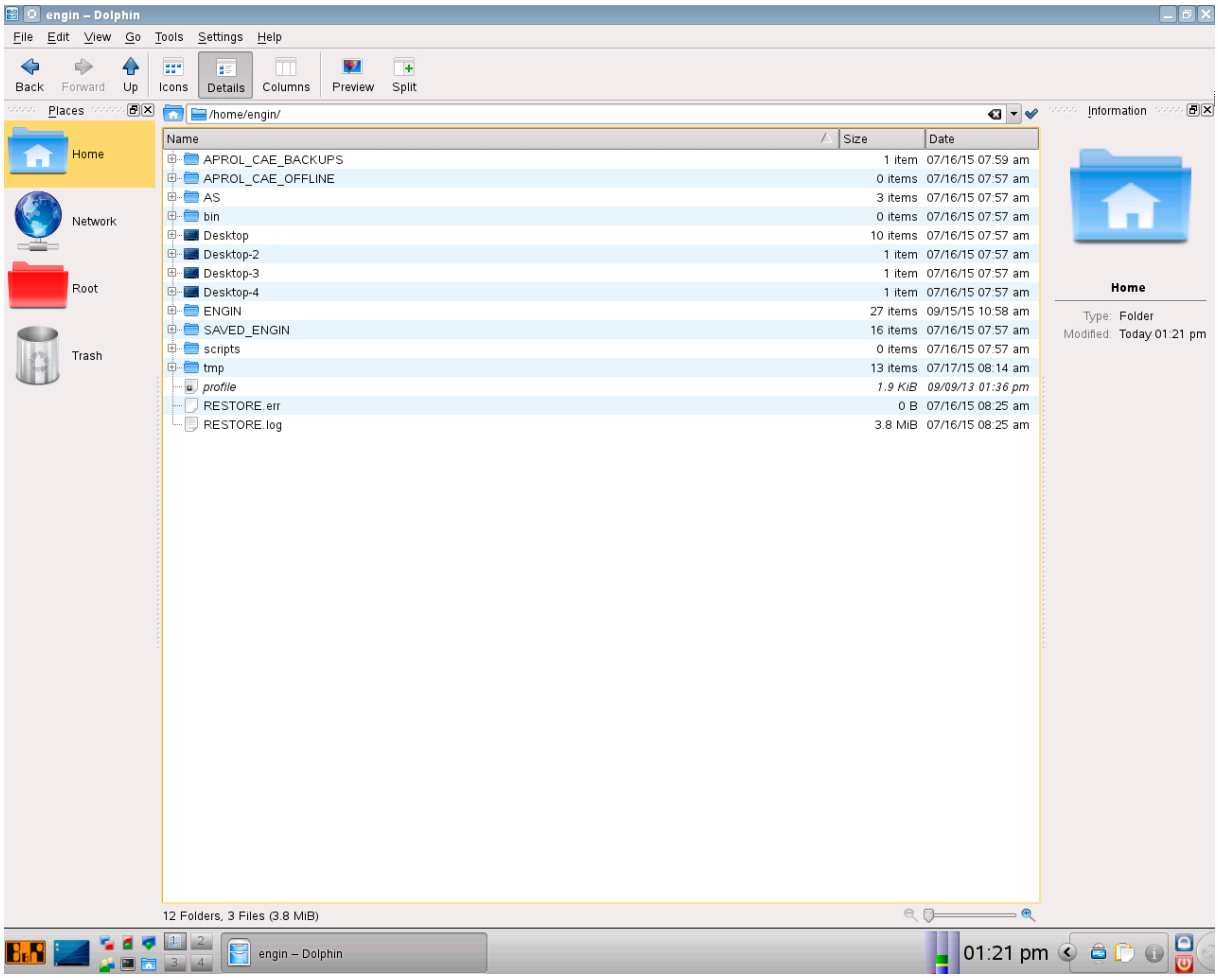


Figure 9: Konqueror for graphical navigation

A console can be used as an alternative. This is then necessary when a graphic user interface is not available (e.g. a pure "server operation" in runlevel 3). Knowledge of the following example is recommended.

7.2.1 Addressing the Directories

Commands for navigating through the directories:

Command shortcuts	Description
cd	Change directory
~/	Option for cd: The home directory of the user
/home/engin	Option for cd: Absolute addressing of the engin directory
../	Option for cd: Relative addressing; one directory up

Command shortcuts	Description
pwd	Outputs the current working directory or current path

A directory change can be done with the addressing mentioned above.

Addressing directories

cd /home/engin/ENGIN	Changes to the specified path
cd	Changes to the user root (e.g. /home/engin/)
cd ..	Changes one directory up
cd ../scripts	Changes one directory up and to the scripts directory from there.
cd ../ ../	Changes two directories up

7.3 Wildcards / Joker

The name of one file or a group of files can be addressed with wildcards.

*	All files in the directory are selected.
?	Any character can be in the place of the wildcard
[a-z]	All small letters from a to z are allowed in place of the wildcard
[!a-c]	No small letters from a to c are allowed in place of the wildcard

Wildcards

ls *[b-e]	Lists all files which have the letters from b to e as last character

8 Basic Command

The shell is an important environment for working in Linux. Although command can be executed in a different manner (mouse click), a shell also offers other possibilities and the usage of self-created scripts.

Linux also provides many programs for the administration of the system. The following chapter shows some commands which are useful for a normal user. No special rights are required to execute the commands.

A typical usage of the program is shown in each example. The behavior of the program can be influenced by using options. The options which are available for a program can normally be found in the **option --help**. Detailed information about the program can be found with the "man" help function (manual page).

8.1 Command Input

A command is normally input in Linux with the following syntax:

<command> <options> <parameters>

The input of options and parameters depends on the respective command.

8.2 Overview of Commands

man Help function (manual page)

A documentation can be opened for most Linux commands by entering "man <command name>".

Opening a man page

man cat	Delivers the description for the cat command

cat Show text file

Command "cat"

cat Datei01	Datei01 is output on the console

grep Search for text in file (grep plc.h *.c)

egrep Search the content of files in a directory (egrep -r search string *.c)

find Search for a file in directory structure

locate Search for a file in the system. This file is searched for in a system database. This is actualized automatically once every day. All of the files and programs which the system uses during the day end up in this database.
It is quicker than "find", but only works with indexed files.

diff Compares two files

head Show the head of a file

tail Show the end of a file

cut Show an excerpt of a file (columns, lines)

clear Clear the console display

nl Count the number of lines in a text file

dd The command can be used to clone a hard disk.

sed A stream editor. The command is used to search for a character combination in a text file and edit it. 'sed' is controlled by using options or transferring a control file. The use of regular expressions in the instructions makes it very powerful and difficult to explain. Both examples show a possible usage of the command.

8.3 Standard Input / Output, Pipes

The standard output of the commands described up-to-now was always on the console. This output can also be diverted to somewhere else. On the one side, the output can be written directly to a file, on the other side, the command output can be passed on as input for another command.

prog > myfile The output of the "prog" program will be saved in the "myfile" file. The file is written in doing so.

prog >> myfile The output of the "prog" program will be appended to the "myfile" file.

prog | prog2 The output of the "prog" program will be transferred to the "prog2" via a pipe.

8.4 Text Editor vi

The vi text editor is helpful with work in Linux and APROL, because it can be used in a text console and a graphic console. This editor also has a mild load on resources. The usage can be unaccustomed, because of the following two working modes. This is then no longer a problem after a short period of time.

Command mode

After vi is started, it is in command mode. Different commands can be executed, such as searching for a word, deleting a line, saving the file, closing the editor, etc.

Insert mode

Text can be inserted in the insert mode. You switch to the insert mode with commands such as **<i>, <a>, <o>** and **<insert>** . If you are in insert mode, you can start typing at the cursor position. Pressing **<insert>** switches between insert and replace. Pressing **<Esc>** returns back to the command mode from the insert mode.

Command	Description
vi MyTestFile.txt	Opens the "MyTestFile.txt" file
<i>, <a>, <o> and <insert>	Changes to the insert mode
<Esc>	Change back to the command mode
:w	Save the file
:q	Close vi

Command	Description
:4	Moves the cursor to the beginning of line 4
:$	Moves the cursor to the end of the file
/asdf	Search for the text "asdf" in the file
dd	Deletes the entire line at the cursor position

Open file, edit, and save

Change to one of your user directories (e.g. /home/engin/tmp). Create a new text file there. Insert several lines of text into the file and save the file.

Close vi and re-open the file to check the previous steps.

9 User Management

As already mentioned, a user must always be logged into Linux. The system therefore knows who is involved. Each user has certain rights which are assigned by the system administrator. System files can therefore no be changed by a normal user. This is important, so that files cannot be changed or deleted by mistake. It is also not possible to access files from other users. The following rule applies:

<Basically, everything that is not explicitly allowed is forbidden>

User groups are used to organize the access of the different users. Several users can be collected together and files and groups deregulated individually. This simplifies the security management and access rights.

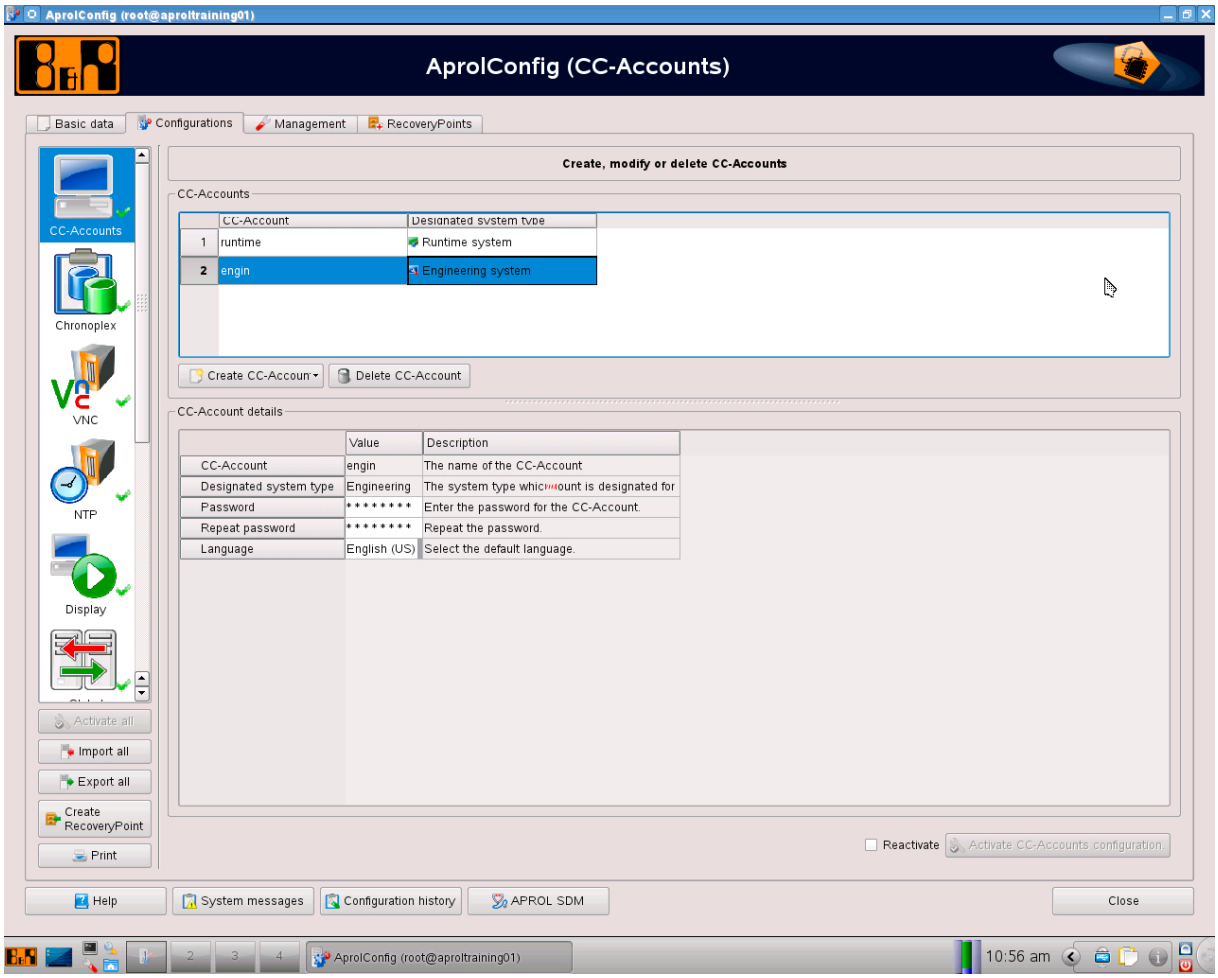
9.1 General Information

Linux is a multi-user system. Several users can be logged into the system at the same time. Each user must identify themselves, so that the data does not get mixed up. Each user is a member of at least one group.

The users and groups can only be changed with root rights.

9.2 User Management in APROL

The user management in APROL is done with the "**AprolConfig "**tool. A user for engine, runtime, gateway, or operator can be created in the dialog. Details about this are in the APROL documentation and in the training manual "TM810 Setup & Recovery".



9.3 User Management with YaST2

Linux standard installations (without APROL) use the program YaST2 (Yet another Setup Tool), menu item **"Security and Users",** to manage the users and assigned groups.

The configuration should not be changed using YaST2 when APROL is being used. The user management should only be done with **AprolConfig**.

10 File Rights

10.1 General Information

As already mentioned, Linux is a multi-user system. Files and directories must be protected from unauthorized access because of this.

Files and directories are assigned to owners and can be assigned read, write, and execute rights.

10.2 Types of Users

Linux differentiates three different groups for file access rights

user	Owner of the file; abbreviated with u
group	All users assigned to a certain group. Abbreviated with g.
other	All other users in the system. Abbreviated with o.

10.3 Access Rights of a Group

The file access rights can be set for each of the above mentioned groups.

r	Read	= access right read
w	Write	= access right write
x	execute	= access right execute

The following abbreviations also exist:

d	directory	= Identifier for a directory
l	link	= Identifier for a reference/ link
t	sticky bit	= only allowed to be deleted by owner

10.4 Showing the Access Rights

The current file access rights can be queried with the "**l file name**" command.

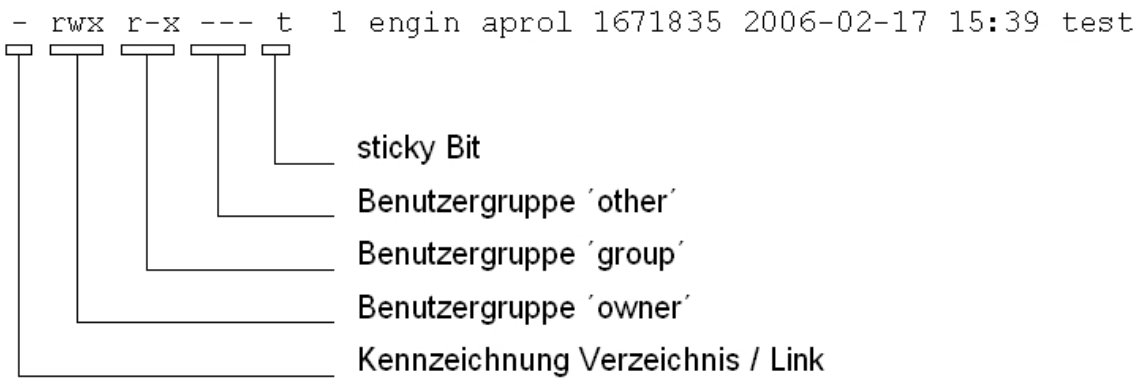


Figure 10: Display access rights

10.5 Command to Change the Access Rights

File access rights can only be changed by the owner or "root".

The following commands are available to change the rights.

Command	Description
chown	Changes the owner of a file
chgrp	Exchange the group affiliation of a file
chmod	Changes the access rights of a file

Changing access rights

chown root test	The owner of the "test" file is changed to the user "root"
chgrp aprol test	The group affiliation of the "test" file is changed to the group "aprol"
chown aprolsys.aprol test	Combination of "chown" and "chgrp" Changes the owner of the "test" file to "aprolsys" and the group affiliation to "aprol".
chmod 777 test	Changes the access rights of all users to rwx (read, write, executable) (see subchapter "Octal Display of Access Rights")
chmod 740 test	The owner gets the access rights r, w, x. The group gets the access right r. All other users have no access rights. The right was transferred in octal format.

10.5.1 Octal Display of the Access Rights

In the example, the access rights were changed by specifying them in octal format. The following table shows how the value is calculated.

	user			group			other		
Possible rights	r	w	x	r	w	x	r	w	x
Conventional representation	r	w	x	r	-	x	-	-	x
Binary representation	1	1	1	1	0	1	0	0	1
Octal representation	7			5			1		

Table 1: Display formats

11 Processes

11.1 General Information

- Processes are programs which have been started.
- Each process has an owner.
- Each process has a parent process and is the child of such a process. (we speak of "parent" and "child" in English)
- A process can be in a sleeping, active, or halted state
- The processes are numbers with a unique PID (Process Identification Number).

11.2 Command for Process Management

ps List of the running processes (process status)

Command "ps"

ps -ax	A list of all of the system processes is output
ps -aux	A list of all of the system processes is output together with the information which user is responsible for the corresponding process.
ps -aux | grep losys	All "losys" processes are output with their user.

kill Sends a signal to a process
The process is requested to terminate gracefully if a signal number is not given.

Command "kill"

kill 12345	PID 12345 will be terminated
killall CaeManager	All CaeManagers will be terminated.
kill -9 12345	Termination of PID 12345 will be forced (even zombie processes - child processes which have lost their parent - are terminated)

12 Network / Ethernet

12.1 Configuring the network card

The Ethernet interface can be configured with YaST2. A new card can be configured or an existing card can be changed with the **<Network Devices>/<Network Card>** menu item.

The illustration shows the network address configuration. It is possible to select between automatic address assignment via DHCP or a static address.

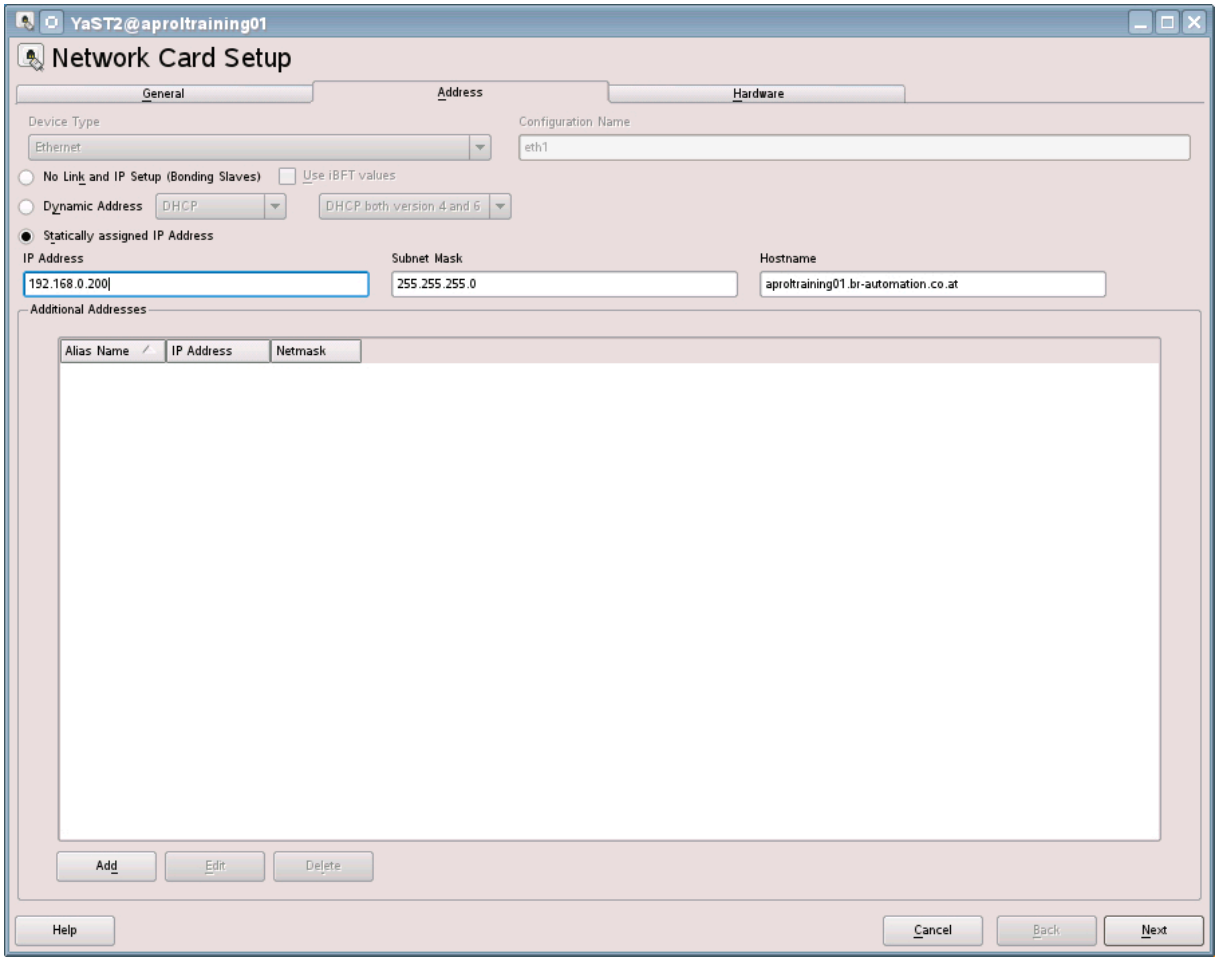


Figure 11: Configuration Network

The illustration shows the routing configuration. If an Ethernet address which is unknown to the system is addressed, the data package is routed to the standard gateway.

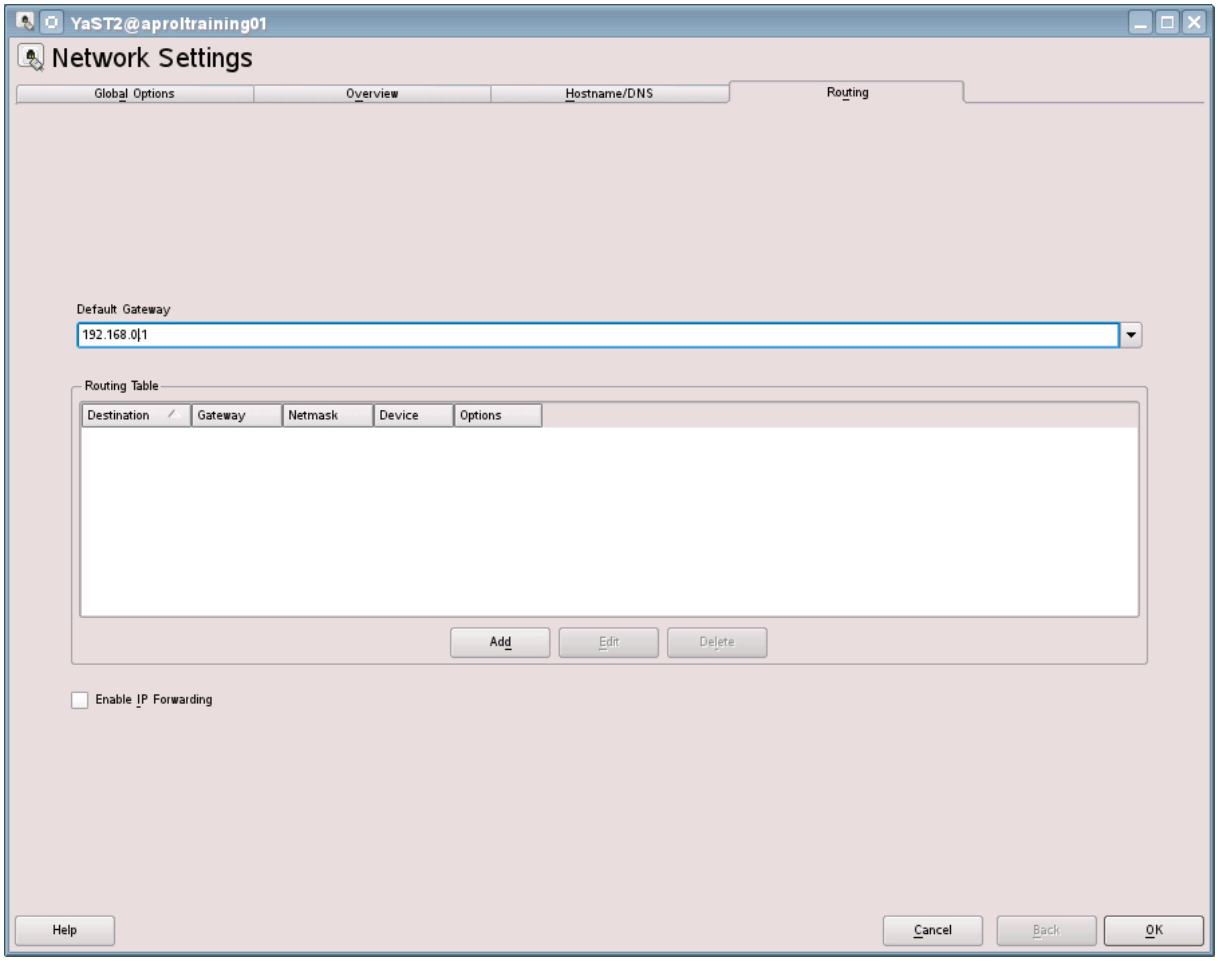


Figure 12: Configuration Network Gateway

12.2 Identification of the network interface

Many network cards may be installed in a computer. The corresponding interface can be made to blink, in order to find out which configured interface corresponds to which physical interface.

In the graphical interface: YaST

The illustration shows the network interface configuration dialog. The functionality for blinking the interface LEDs can be found in the "Hardware" tab.

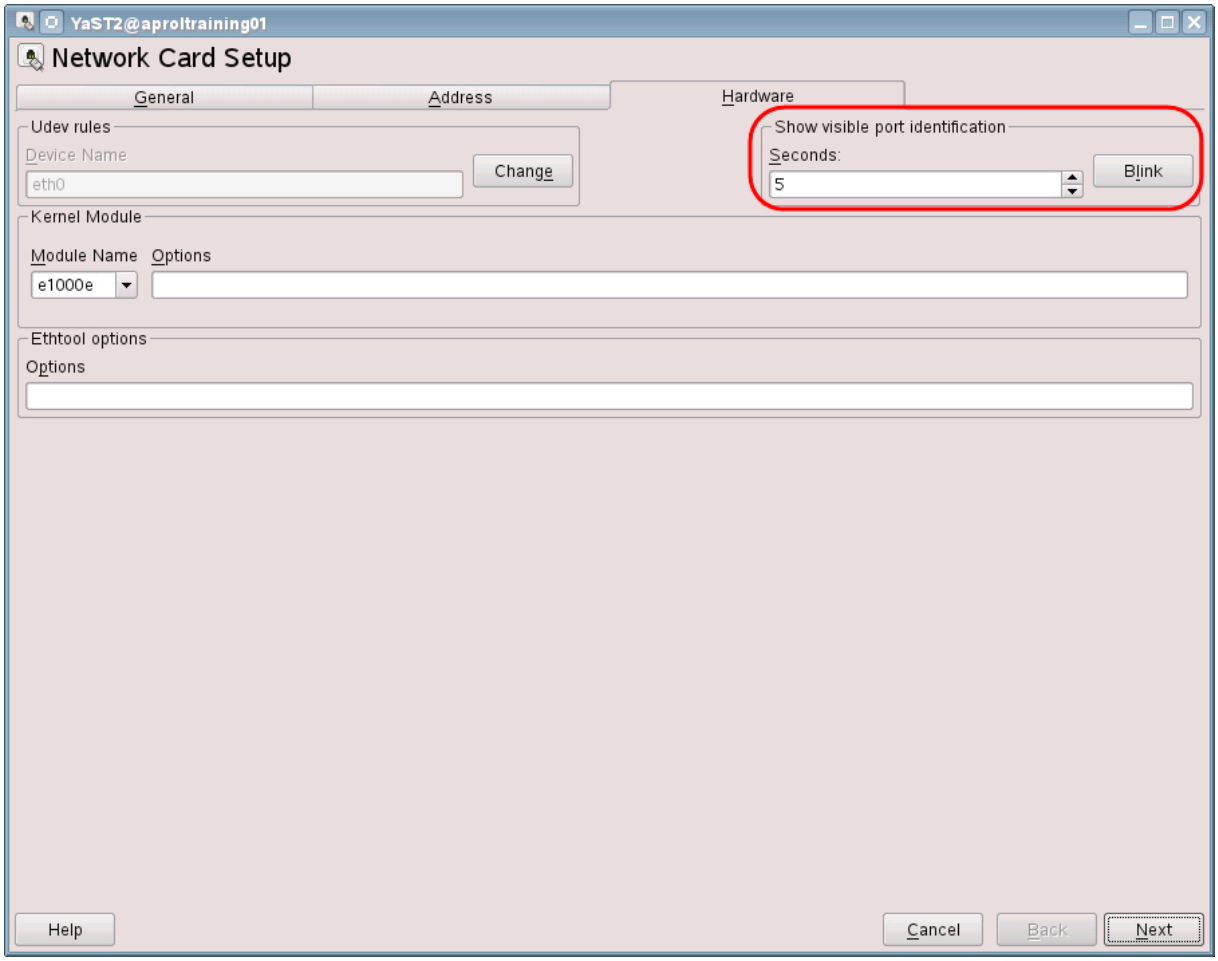


Figure 13: Blinking the network interface

In the text console: ethtool

As an alternative to the graphical user interface, the **ethtool** command can be executed in a text console.

If the user is root, the command can be executed immediately. All other users must also specify the absolute path (/sbin/ethtool).

ethtool -p eth0 Makes the eth0 interface blink

12.3 Tools for network management and diagnosis

ping Checking if a certain participant (host) can be reached in an IP network.

Command "ping"

ping 192.168.75.1

The request is sent to the specified address. The duration of the query will be shown if the address can be reached in the network.

ping -c 10 -s 1000 192.168.75.1

This option is a simple way to check if packages are lost. Packages of the size **s** (specifies the number of data bytes which should be sent) are sent 10 times (**c** stands for "count") and the number of lost packages is calculated.

ifconfig Display of the current network card configuration

If the command is started without any additional arguments, the current configuration of all network cards will be output.

If the user is root, the command can be executed immediately. All other users must also specify the absolute path (/sbin/ethtool).

traceroute The route from the computer to the target computer is displayed.

13 Working with Data Storage Media

13.1 General Information

Media such as CD-ROM, DVD or other exchangeable media are integrated in the directory structure in Linux. This is normally done automatically after the media is inserted and the computer is in runlevel 5. Automatically integrated devices are shown in the "/media" directory. If the computer is not in runlevel 5, the media (this may also be an ISO file) must be integrated manually. The command for integrating a medium is "mount".

13.2 Manual Integration of Media

mount <device> <mount point>	Integrate device manually

If the "mount" command is started without arguments, all of the integrated file systems will be shown.

13.2.1 General Integration of Devices

"mount" of a USB stick

mount /dev/sdb /mnt	The device **sdb** is connected with the existing /mnt directory.

The name of the media device files can be detected in the F10 console by inserting the media.

13.2.2 Integrating an ISO Image

An ISO image can be copied to the hard disk, for example. This file can be integrated into the file system. There is no difference when accessing this file to that of accessing the CD-ROM. Only the "mount" command and its options varies.

Integrating ISO images

mount -t iso9660 -o loop APROL-AutoYaST-DVD-V4.0-110-SLES11.iso /mnt

The specified ISO image is connected with the existing /mnt directory.

13.2.3 Integrating Linux Network Directories

The Linux service "Network File System" (NFS) is for direct access to data on other computers.

The following requirements must be met to access remote computers via NFS.

- The computer must be reachable in the network.
- The NFS server must be running on the remote computer.
- The user must have the right to access the remote computer.

"mount nfs"

mount -t nfs rechner_ext:/data /mnt

The /data directory on the remote computer with the name "rechner_ext" is connected with the existing /mnt directory.

13.3 Command for Removing Media

umount (or un-mount)	Remove media from the Linux file system

14 Backing up Data

tar Create data archive

The universal Linux tool for backing up data is the "tar" command. The command can unify several files or directories in one archive.

Command "tar"

tar cfvz Archiv.tgz ENGIN	The directory ENGIN and all of its data will be summarized in the archive with the name Archiv.tgz. (c = create)
tar xfvz engin_1234.tgz	The archive with the name eingin_1234.tgz will be unpacked with all data in the current directory. (x = extract)
tar tfzv engin_1234.tgz	The archive will be read and shown, but not unpacked. (t = tail)

15 Collection of Helpful Commands

There are many helpful commands in the following table with short descriptions. This table does not contain all of the Linux commands. If you do not find what you require (command), we would like to point out the Linux manuals in book stores and the available online information.

Command	Description	Example
at	Executes the command at a certain time	reboot at 18:00
bg	Let specified processes run in the background	bg %12 (process 12 in the background)
boot	Booting the computer	Only system administrator
cal	Calender output	cal 2000 = show the year 2000
cancel	Deletes the started print job with lp	Cancel laser -123 = kill print job no. 123
cat	Concatenation, merging and display of files	cat 1.txt 2.txt > 3.txt = output 1 & 2 in 3 cat > neu.txt = create new file and close with STRG + D in the edit mode
cd	Change directory	cd = change home directory cd ..= one directory up cd ~/hans=home directory of hans cd /home/hans= absolute
chgrp	Changes the group affiliation	chgrp "group name" file/directory chgrp -R > recursive
chmod	Change access rights	Method1: chmod ugo +/-/= file name u=user, g=group, o=owner, +=add, -=eliminate, ==set absolute, r=read, w=write, x=executable Methode2:chmod 750 file name (octal number) r=4,w=2,x=1, 1.number=owner, 2.number=group, 3.number=other owner all rights&group read only: U=4+2+1, G=4+0+0, O=0+0+0 also chmod 740 file name
clear	Deletes the current screen content	
cmp	Compares file content	Cmp txt1 text2, the different lines are displayed
cp	Copying files	cp -i old file new file, cp -i file1 file2…directory, cp -ir source directory target directory -i=ask if file exists, -r=recursive (entire directory tree)
cut	Cuts columns from files or character strings	cut -d: -f0,4,5\ (etc/passwd > columns 1,5&6 are output to the screen -d=delimiter, -f=field number, -c=character position

Collection of Helpful Commands

Command	Description	Example
date	Shows the date (only changeable by SU)	date +"%d.%m.%y" > e.g. 30.08.06
df	Shows the available hard disk space (disk free)	df -k > free space in 1KB blocks df -h Human readable (in M or G)
du	Shows the used hard disk space	
Echo	Outputs character strings to the screen	echo "Hello world", echo $PATH > shows the content of the variable (environment)
env	Shows the set variable	Listing of environment variables with the corresponding content (values)
export	Variables will be exported and are then valid for subprograms	
fc	Repeat/display of entered commands	fc -l = show the last 10 commands -l=the last 10 commands, r -n repeat the nth. command
fg	Executes process in the foreground	fg %23 > Process no. 23 in the foreground
file	Tries to ascertain the data type	file 1.txt > result: ASCII text
find	Search for files with the specified criteria	find -name *.log > listing of all *.log files in relation to the directory where the command was started.
ftp	File transfer to and from remote computers	ftp 192.168.75.168> Login: engin> password:.engin eventual cd /directory > mget *.txt > all *.txt files are transferred (with query) . get FILENAME=download file put FILENAME=Upload, mput-mget=several bin=switch to binary m ascii=switch to ASCII mode bye,quit=exit
grep	Search in files for certain strings or criteria	grep "test" name.txt > all of the lines in the document name.txt which contain test will be shown.
halt	Halt the system in order to switch it off	Can only be done by SU
head	Shows the first 10 lines of a file	head -6 name.txt > the first 6 lines will be shown
host name	Shows the computer name	
kill	Terminates a process securely	kill -9 1234 > process 1234 will be terminated -1=ends terminal connection, -2=like STRG+C(cancel), -3=cancel with core dump,-9=absolute process termination, -17=stops a process, -18=stops a foreground process(STRG+Z)

Command	Description	Example
ln	Link to a file or directory	ln -s Original directory Link directory -s=symbolic link (can be set on any partition), if -s is not specified, a local link exists, which can only be set to the current local partition
logout	Log out of a shell	
lp	Creates a print job for the lp spooler	lp -dPRINTERNAME FILENAME -d=Destination(target)
lpq	Shows the print jobs created with lpr	
lpr	Creates a print job for the lpr spooler	
lprm	Deletes the print jobs started with the lpr	lprm 23 > Delete print job 23
lpstat	Shows all print jobs started with lp	
ls	Shows directory content	ls-l >show all files with attributes (owner, rights, etc.) -a=all(also beginning with .), -b=binary, -d= only directory no content, -l=with attributes,-R =recursive with all subdirectories, -s= show the bolocks(512bytes), -t=sort by time
mail	Send mail	mail user@computer
md5sum	Determine the checksum of a file	md5sum <file1>
mesg	Allow/forbid other user messages	mesg y > allow, mesg n > forbid
mkdir	Create directory	mkdir -p tmp/test > Directory tmp and subdirectory. Test will be created
more	Shows the file content page by page	
mount	Integrate device into the file system	mount /dev/hdc1 /mnt disk mounted to /mnt
mv	Moving files	mv -i *.txt /home/engin/tmp > move all TXT files in the specified directory -i=asks if existing files should be overwritten
passwd	Change the password	passwd USERNAME > Demands entering a new password
ping	Is the addressed computer available in the network?	ping 192.168.75.168
ps	Displays the current processes	ps aux > all processes with user information
pwd	Show current directory	With absolute path
rcp	Copies files to a remote computer	Only possible via TCP/IP
rlogin	Log in to a remote computer	Only possible via TCP/IP rlogin -l USERNAME IP-ADDRESS

Command	Description	Example
rm	Delete files permanently	-f =also write-protected files without query, -i=confirm deletion with y, -r= recursive (including all subdirectories)
rmdir	Delete empty directory	
set	Shows the currently set variables	
shutdown	Shutting the computer down	Can only be done by the SU
sleep	Introduce a waiting time/pause	In seconds, sleep 120 > 2 wait 2 minutes
sort	Sort file content of strings according to the specified criteria	sort -t: -k5.1,5.4Vetc/passwd >passwd will be output sorted by the first 4 letters in the name in ascending order -f=Capital & small letters are equivalent, -t =separator, -n=numerical values are sorted, -k=Position, -r=reverse sort
tail	Shows the last lines of a file	-n=number of lines(10 Standard), -f= the new lines are update when the file is edited
tar	Saving data in an archive	tar -cvzf name of the directory > Directory and entire contents will be written to file Name.tgz and zipped. (create verbose zip file) x= extract, t=tail
telnet	Log in to a remote computer	telnet 192.168.75.168 > Login with password
touch	Update the file data or create a new file	Touch 1.txt 2.txt > Create file 1.txt and 2.txt
tty	Shows the current terminal type	
man	Manual Page	A detailed description about each command can be opened. man telnet > delivers the description for telnet
vi	Screen-oriented editor	vi 1.txt > 1.txt open (or create) and edit. Basic commands: Einfg > Insert mode (re.press > replace mode, Esc back) :w=save, :q=quit, .q!=quit untouched, /=search, :$=End of file, :line number= jump to line

Space for own notes

Command	Description	Example

Command	Description	Example

Collection of Helpful Commands

Command	Description	Example

16 APROL File Structure

The following system can be created in APROL with the **AprolConfig** program:

- Engineering System
- Runtime system
- Operator system
- Gateway system

Each system shown here is created as a Linux user and has a directory in the "/home/" directory. The directory structure differs according to the type of system. The following chapter explains the organization of the individual systems.

16.1 File Structure Engineering Environment

A data required for a project is in the "engineering environment". Everything can be found here, beginning with the engineering database, going on to created files (e.g. C code), and up to HTML documentation.

The basic directory is "/home/<engineering user>/".

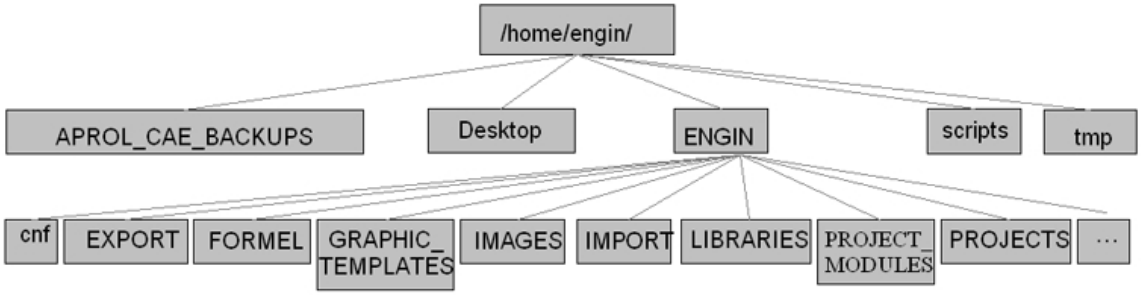


Figure 14: Directory structure ENGINEERING SYSTEM

This illustration is the first overview. The important files in the subdirectories are described in the following.

The most important directory is the "ENGIN" directory. All of the project files are in this directory.

Directory	Description
APROL_CAE_BACKUPS	This is the standard directory for engineering backups. This is the standard path for a project recovery in the APROL CaeManager.
Desktop	Data for the desktop (as in the Windows system)
ENGIN	All of the engineering data of the APROL project engineering. Any "external" data (HTML, configurations, AS libraries, etc.) can be found here.
scripts	Directory for user-specific scripts (e.g. CustomerPostBuildScript, etc.)

Directory	Description
tmp	Directory for temporary data

16.1.1 The ENGIN Directory

This directory contains a large number of subdirectories, in which important engineering data is stored. The engineering database is located directly in the "ENGIN" directory and is composed of the "caedb.link" and "caedb.tree" which are used by the CaeManager. Even so, it is not possible to store all of the necessary data in the database. This is the reason why several subdirectories were created in the "ENGIN" path, in order to store additional data. Furthermore, data is written from the database to the file system.by compilation and generation activities, and are thus made available for a download.

Listing of the first level in ENGIN:

Subdirectory	Description
cnf	There are external configurations for applications in cnf, such as Alarmzeile.xml for configuring the alarm line of the DisplayCenter. This data is integrated during a generation (devil) and made available for download.
CUSTOMER_GLOBAL	Project-dependent files can be stored here and will also be saved during a database backup (CaeBackup).
EXCHANGE	Data which is exported from the engineering database using the CaeManager is normally saved here.
FORMEL	External formulas which may be necessary can be found here (scripts). These are also made available for download during the generation process.
GRAPHIC_TEMPLATES	Storage location for graphic templates which can be created with the DisplayEditor.
IMAGES	All images and background images are normally stored here.
LIBRARIES	All of the translated libraries in the CaeManager write their data to this directory. (during compilation)
PROJECT_MODULES	Directory for the necessary data in case AS customer-specific libraries are integrated
PROJECTS	Storage location for the projects created in the CaeManager. A directory "Project name.pgp" (see AS) is created with subdirectories. The entire project and all created data (also for the download) is here.
SOUND	Sound files which may be used (e.g. WAV) are stored here. These are made available for a download after the generation process.
TEXT	Text files which may be used are stored here.

The following illustration shows the ENGIN subdirectories and the area project data:

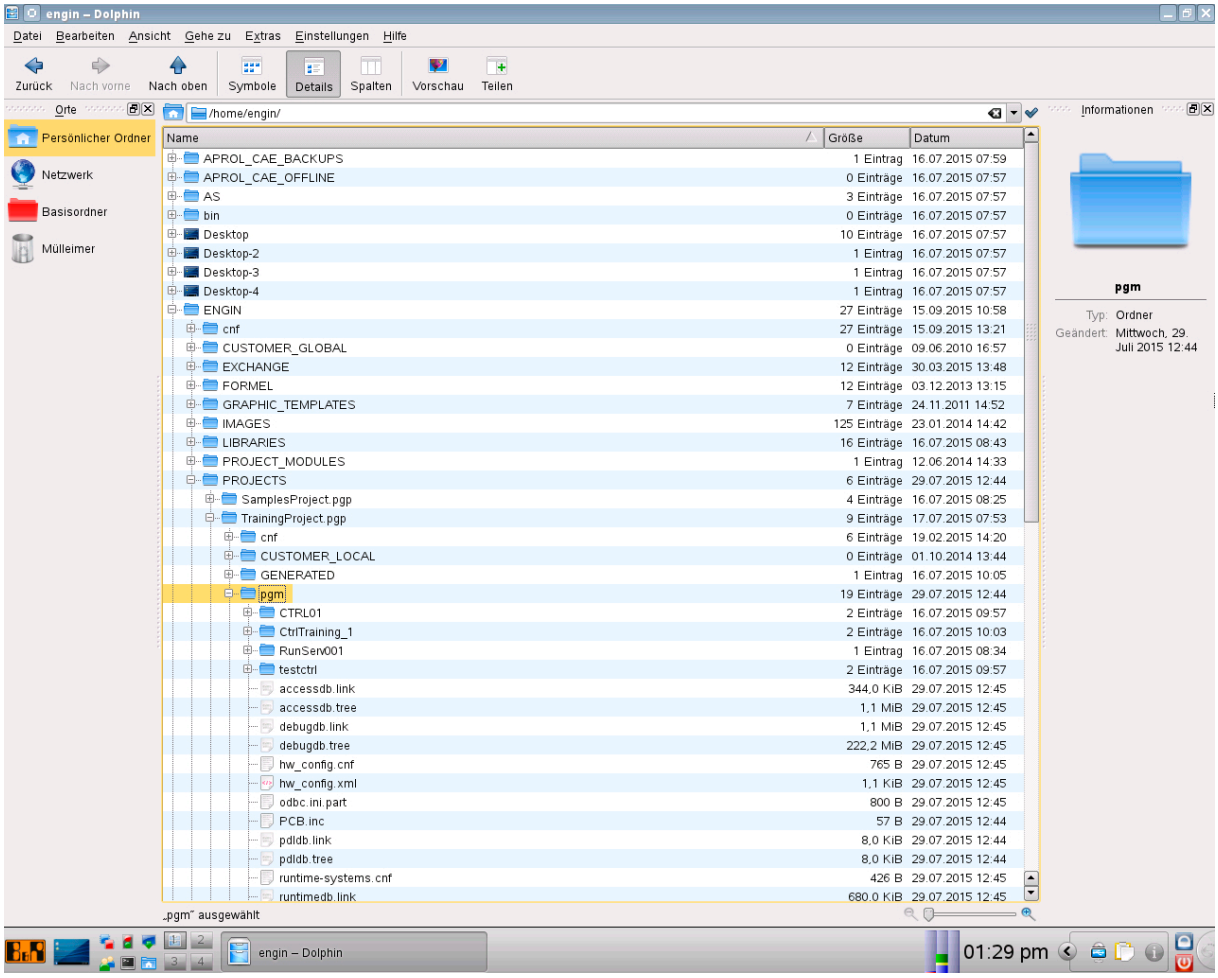


Figure 15: Data structure ENGIN

16.1.2 Structure of the Created Project Data (Project.pgp)

Data is written to the engineering environment file system when compiling and generating in the CAE engineering. This can be found in the area "ENGIN/PROJECTS". A "Project name.pgp" directory is created for each project which is created in the database.

These directories can be used for analysis of and searching for data, because they contain all of the created project data. For example, all function chart C files and tasks can be found here. All physical view entries can be viewed and special configurations (see cnf) can also be found here.

Additionally, the dbk directory which depicts the variable image of the controller is important in order to be able to carry out a new organization of the variable image (deletion > generation).

The subdirectories of a created project are explained in the following.

Directory	Description
cnf	Configurations for different applications such as the DisplayCenter (IMAGES, Background images, etc.), KDE (specific settings), GLOBAL (rhosts, ProfibusFMS configuration), etc.

Directory	Description
CUSTOMER_LOCAL	Project-specific files can be stored here and will also be saved during a database backup (CaeBackup).
GENERATED	All data for all types of downloads can be found in this directory. The GENERATED directory is the source for the DownloadManager. The substructure is the physical view of the controller. On the other hand, all created data such as databases (runtimedb), all tasks, and system files for the different controllers are here (physical view CC/cnf/ControllerLoader/ Ortswelt Controller/files.br).
pgm	All of the files created during compilation and generation (C files, HW addressing, databases, etc.) can be found in the pgm directory. The subdirectories are those of the physical views of the CC and controllers.
WEB	External documentations, intervention texts (alarm system), reports, and customer-specific reports must be stored in the WEB directory. A further structure of this directory is made up of these points (country code first).

16.2 File Structure Runtime Environment

All of the necessary data, databases, configurations, etc. are downloaded with the DownloadManager from the engineering to the runtime environment. A defined structure of the directories also results from this.

The basic directory is "/home/<runtime user>":

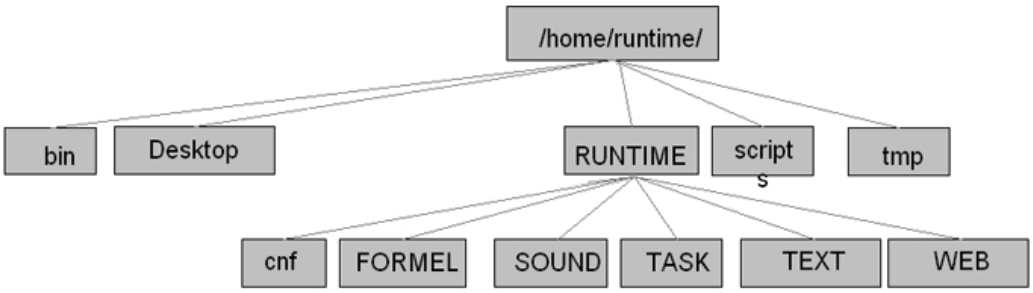


Figure 16: Overview directory structure runtime environment

The most interesting directory is the "RUNTIME" directory. For example, databases and the remanent Iosys data are in the "/home/<runtime user>/RUNTIME". These files are important, because they are, amongst others, the values which are currently entered in the DisplayCenter.

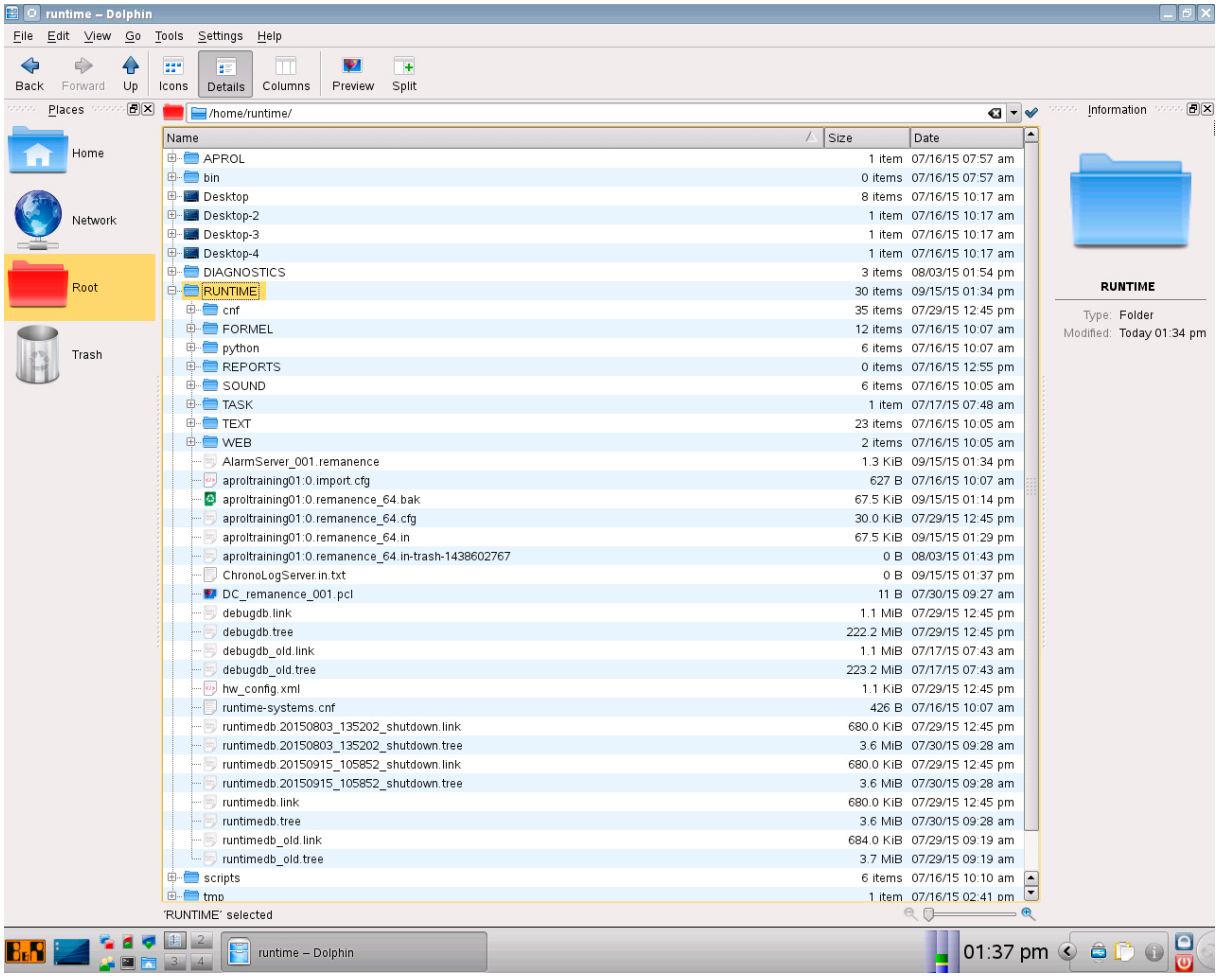


Figure 17: Structure of the RUNTIME directory

The following information can be drawn from the subdirectories:

Directory	Description
cnf	This directory contains all configurations of different applications such as the DisplayCenter, alarm report, ControllerManager, etc.
FORMEL	All formulas which are called and executed by the APROL formula interpreter are stored here. (scripts)
python	Global Python modules and library type definitions can be found here.
SOUND	Sound files for the runtime environment, e.g. for the alarm system
TASK	The control computer tasks are stored here. Executable file (files) for the runtime environment)
TEXT	Directory for text and HTML files which are necessary for the DisplayCenter and runtime environment.
WEB	All of the web-based files (HTML) for intervention texts (alarm system), documentation, and customer-specific reports can be found here.

16.3 File Structure Operator Environment

The operator environment is structured similarly to the runtime environment. Only the different background processes and driver are missing (Iosys, AnslDriver, etc.).

17 Summary

When you have reached this chapter, you have made a successful start in working with Linux operating system. The strengths of Linux will become more obvious as you continue working with the system. Furthermore, you should now also have a sufficient overview of the Linux file system and APROL.

You should be able to work with APROL and the SUSE Linux Enterprise Server (SLES) comfortably and without any problems.

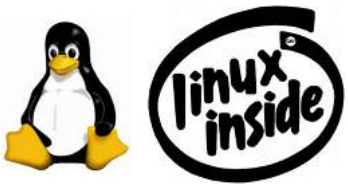


Figure 18: Linux identification

Seminars and training modules

The Automation Academy provides targeted training courses for our customers as well as our own employees.
At the Automation Academy, you'll develop the skills you need in no time!
Our seminars make it possible for you to improve your knowledge in the field of automation engineering.
Once completed, you will be in a position to implement efficient automation solutions using B&R technology. This will make it possible for you to secure a decisive competitive edge by allowing you and your company to react faster to constantly changing market demands.

Automation Studio seminars and training modules

Programming and configuration	Diagnostics and service
SEM210 – Basics SEM246 – IEC 61131-3 programming language ST* SEM250 – Memory management and data storage SEM410 – Integrated motion control* SEM441 – Motion control: Electronic gears and cams** SEM480 – Hydraulics** SEM1110 – Axis groups and path-controlled movements** SEM510 – Integrated safety technology* SEM540 – Safe motion control*** SEM610 – Integrated visualization*	SEM920 – Diagnostics and service for end users SEM920 – Diagnostics and service with Automation Studio SEM950 – POWERLINK configuration and diagnostics* If you don't happen to find a seminar on our website that suits your needs, keep in mind that we also offer customized seminars that we can set up in coordination with your sales representatives: SEM099 – Individual training day Please visit our website for more information****.****: **www.br-automation.com/academy**

Overview of training modules	
TM210 – Working with Automation Studio TM213 – Automation Runtime TM223 – Automation Studio Diagnostics TM230 – Structured Software Development TM240 – Ladder Diagram (LD) TM241 – Function Block Diagram (FBD) TM242 – Sequential Function Chart (SFC) TM246 – Structured Text (ST) TM250 – Memory Management and Data Storage TM400 – Introduction to Motion Control TM410 – Working with Integrated Motion Control TM440 – Motion Control: Basic Functions TM441 – Motion control: Electronic gears and cams TM1110 – Integrated Motion Control (Axis Groups) TM1111 – Integrated Motion Control (Path Controlled Movements) TM450 – Motion Control Concept and Configuration TM460 – Initial Commissioning of Motors TM500 – Introduction to Integrated Safety TM510 – Working with SafeDESIGNER TM540 – Integrated Safe Motion Control	TM600 – Introduction to Visualization TM610 – Working with Integrated Visualization TM630 – Visualization Programming Guide TM640 – Alarm System, Trends and Diagnostics TM670 – Advanced Visual Components TM920 – Diagnostics and service TM923 – Diagnostics and Service with Automation Studio TM950 – POWERLINK Configuration and Diagnostics TM280 – Condition Monitoring for Vibration Measurement TM480 – The Basics of Hydraulics TM481 – Valve-based Hydraulic Drives TM482 – Hydraulic Servo Pump Drives TM490 – Printing Machine Technology In addition to a printed version, our training modules are also available on our website for download as electronic documents (login required): Visit our website for more information: **www.br-automation.com/academy**

Process control seminars and training modules

Process control standard seminars	Process control training modules
SEM841 – Process Control Training: Basic 1 SEM842 – Process Control Training: Basic 2 SEM890 – Advanced Process Control Solutions	TM800 – APROL System Concept TM811 – APROL Runtime System TM812 – APROL Operator Management TM813 – APROL XML Queries and Audit Trail TM830 – APROL Project Engineering TM890 – The Basics of LINUX Visit our website for more information: **www.br-automation.com/academy**

* SEM210 - Basics is a prerequisite for this seminar.
** SEM410 - Integrated motion control is a prerequisite for this seminar.
*** SEM410 - Integrated motion control and SEM510 - Integrated safety technology are prerequisites for this seminar.
**** Our seminars are listed in the Academy\Seminars area of the website.
**** Seminar titles may vary by country. Not all seminars are available in every country.