Installation Manual

DIAB Series 90 - model 00

Updated for D-NIX System 5.2 version 1.2

9703-00

Diab Data AB

Box 2029 S-183 02 TÄBY

SWEDEN

Revision information

Revision information

Revision	Date	Comment
A	870624	First version for D-NIX 5.2 version 1.2

Contents

Contents

INTRODUCTION

Welcome to this manual, describing the installation of the DIAB Series 90 - model 00 computer system. The main purpose of the manual is to guide you through the unpacking of the system, connection of devices, start up of the system, and to set-up the first system configuration.

The manual also gives a description of what possible media to use and of devices attachable to the system. Further there is a section describing how you connect different terminals, printers and modems using suitable cables.

In chapter 3 is shown how to operate the system from the front panel and the help routines available for the system administrator to make management and preventative maintenance easier.

Especially described in detail is the usage of the system commands for your system in particular. This is done with complete examples and lists of different valid parameters.

In addition to this there are also some hints on how you can avoid the most common obstacles during system work.

System documentation

```
9764-00 D-NIX User manual
9703-00 DS90-00 Installation manual
9830-00 System notes
9792-00 System administration DS90-00
9701-00 D-MENU
```

In this manual we refer to other manuals with a two-letter code. The codes are:

UM User Manual
SN System Notes
SA System administration
DM D-MENU

We would very much appreciate having any comments on this manual from you as a user. If you have ideas about how it could be improved, please use the customer comment form at the end of the manual.

Good luck with your new computer system.

Diab Data AB,
Täby, Sweden

Introduction

1. UNPACKING

1.1 Unpacking

The system is delivered in a package according to the figure below.

NOTE! The system has many sensitive components, e.g a Winchester disc unit that cannot stand hard vibrations. Therefore the system must be handled with caution.

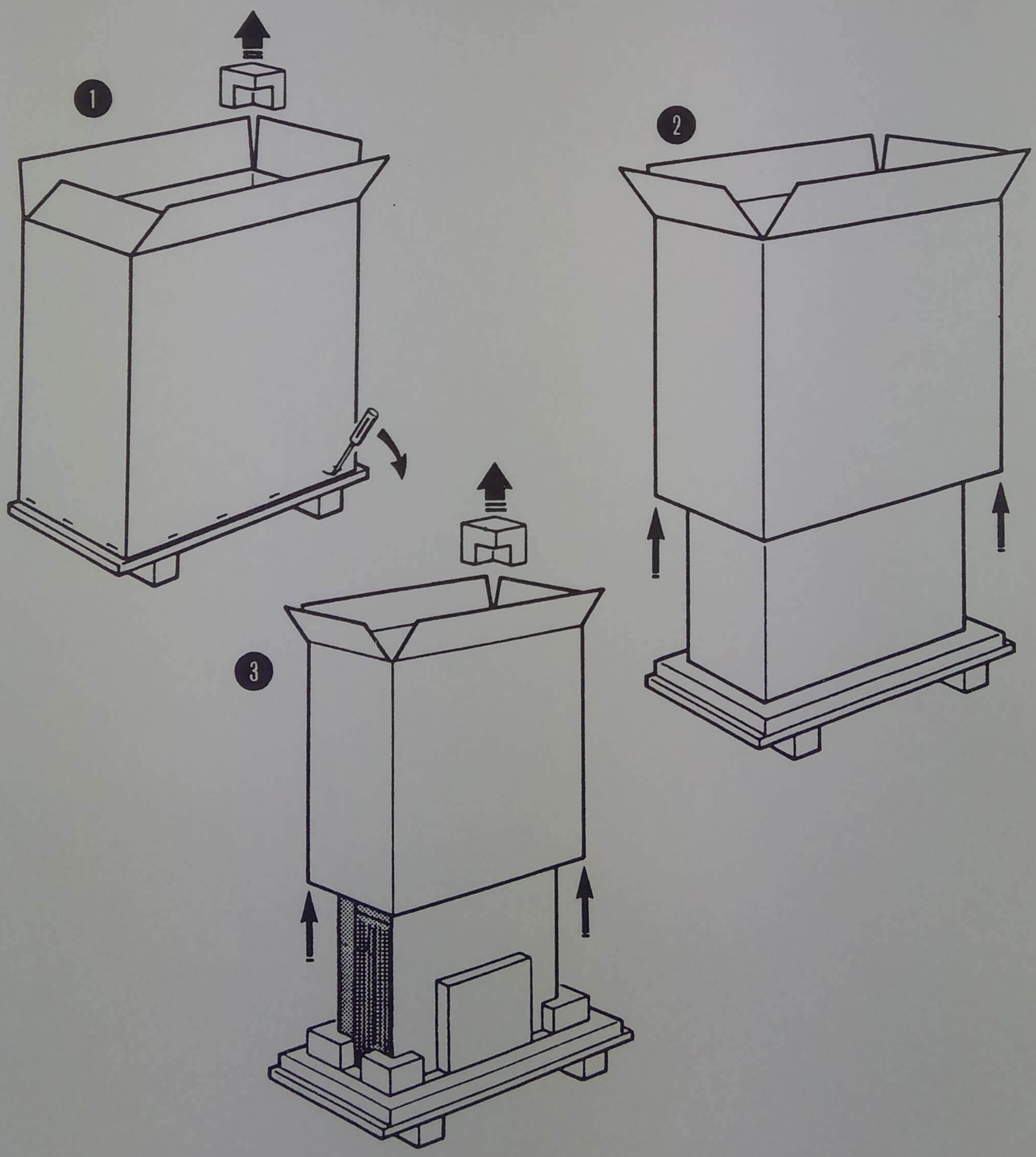
Unpacking the system

Prepare the site for the system setup before the unpacking.

* Open the outer package from the top and remove the foam blocks.

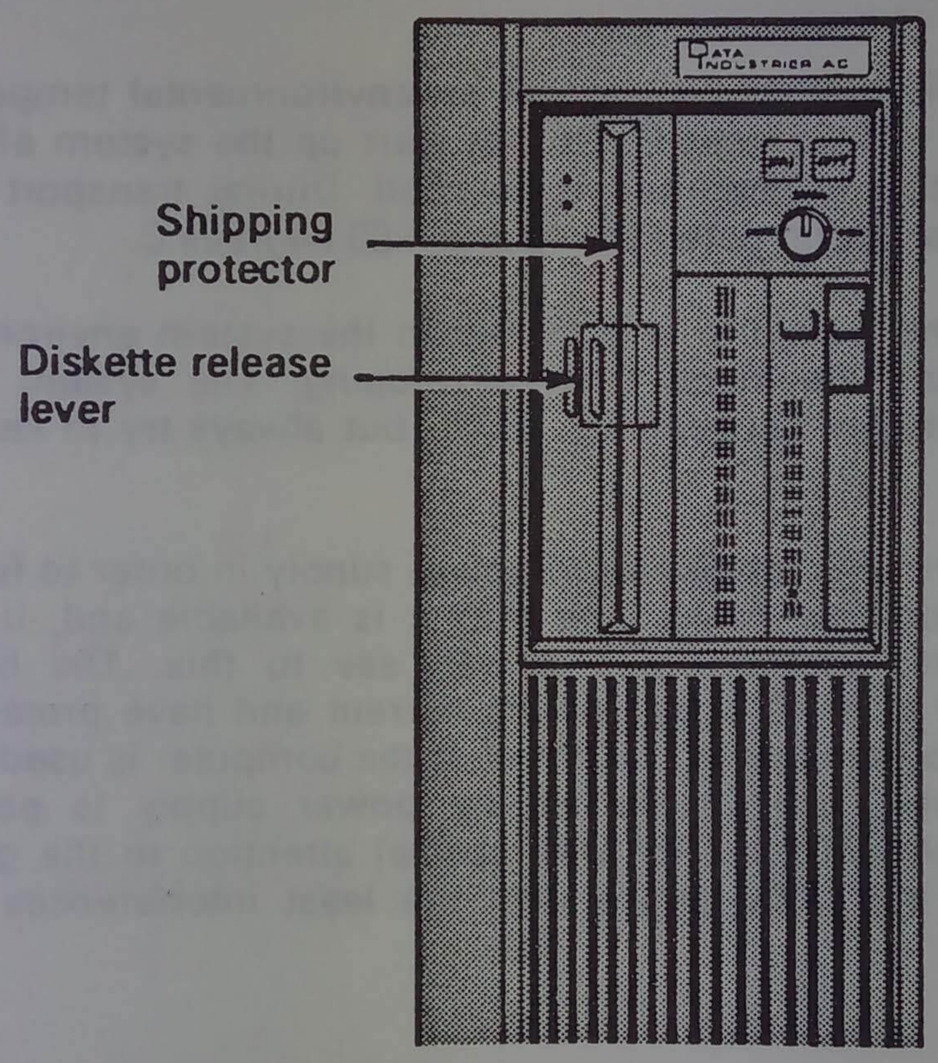
* Loosen the outer package from the pallet at the bottom by removing the staples with a screwdriver.

* Carefully lift the outer package straight up so that the inner package containing the computer remains on the pallet.



* Open the inner package from the top, remove the foam blocks and lift the inner package straight up leaving the computer on the pallet. Observe the keys fastened with tape on the back of the computer.

* Lift off the computer and put it in an upright position on a flat surface.



The 8 inch floppy disk unit has a white cardboard disc inserted as transport protection. This must be removed.

* Press the left button on the floppy disc unit. The cardboard disc can now be removed.

* Remove the cardboard disc.

* Save the cardboard disc. It may be needed for later transport of the system.

1.2 Environment for the system

When you plan the environment of the system, there are some items you ought to consider. Generally the computer finds the same environment agreeable as you do yourself. Check the following list before installing the computer.

* Computer systems always generate heat. Therefore, the system always needs good ventilation. The system takes in the air on the front side and evacuates warm air through the fan on the back. To achieve the appropriate cooling air streams inside the computer, cover plates must be mounted over the expansion card holes on the back of the main unit.

* The system is built to function at an environmental temperature between +10 - +40 degrees C. Do not start up the system after a transport before this temperature is reached. During transport or storage the system can stand between -40 and +60 degrees C.

* Verify that the humidity is sufficient in the system environment. This prevents static electricity from developing. The system is built to handle a humidity between 20 - 80 % but always try to keep it above 50 %.

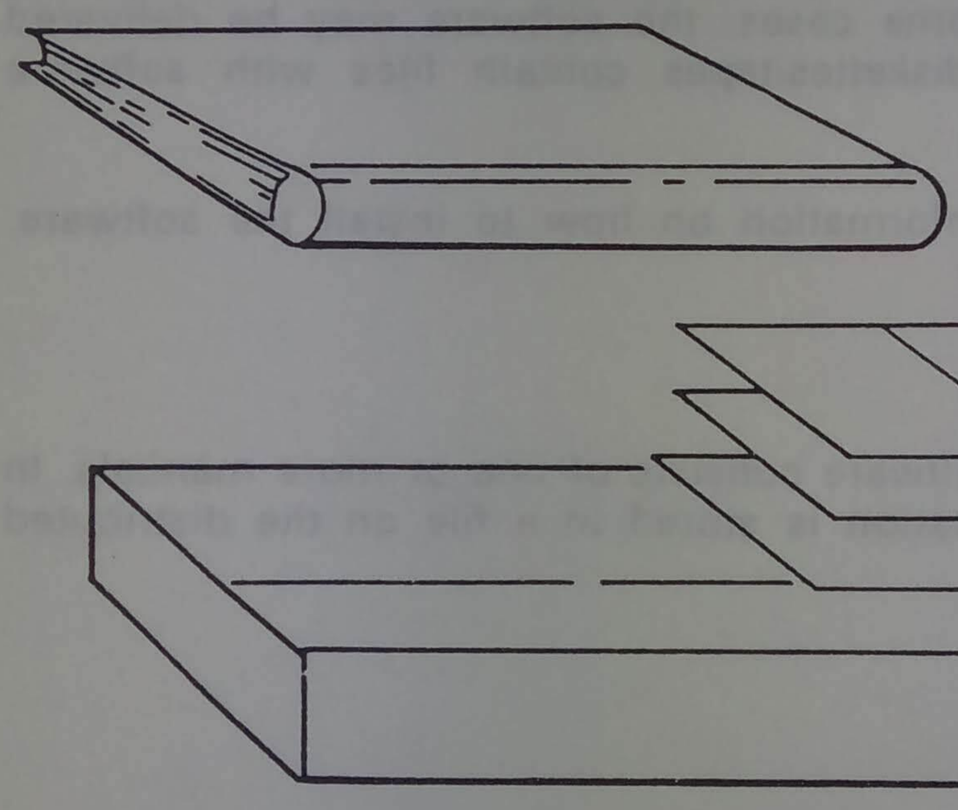
* Computer systems require good power supply in order to function without any problems. Make sure that it is available and, if the installation requires uninterrupted power, see to this. The line voltage shall be 220 V +/-10 % alternating current and have protective earth. Maximum power required is 300 W. If the computer is used in an environment where interference in the power supply is possible, the connection should be done with special attention to the ground connection and the 220V phase with the least interferences should be used.

1.3 Accessory packages at delivery

The system is delivered with a number of accessory packages, all depending on what has been ordered. The contents may be as follows:

* System diskettes with software, release notice and documentation.
* Diskettes with additional ordered software.
* Ordered cables for terminals and printers.

Ordered software and documentation are delivered in packages as below:



1.4 Software

The system is delivered with the basic system package installed on the internal disk unit. Each software package consists of the following:

* Media with software. Here may also be found files that contain information and documentation to the software.

* Documentation

* Release notice

* Licence agreement

* Updating agreement form

Media with software

The software is delivered on diskettes, tape casette or magnetic tape in "tar"-format. In some cases, the software may be delivered in "mount"-format. The diskettes/tapes contain files with software information.

The Release Notice has information on how to install the software.

Documentation

The documentation of a software consists of one or more manuals. In some cases, the documentation is stored in a file on the distributed diskette/tape.

Chapter 1 Unpacking

Release Notice

Release Notice is the name of the folder that comes with all software and with all new versions of software.

In the Release Notice the installation of the software is described. There are also information about differences in and additions to earlier versions. It is of great importance that the system administrator passes this information on to the users of the system.

Licence Agreement

A licence agreement comes with each software delivery. As soon as the customer has returned the signed agreement form to the vendor, the software may be used by the customer.

Updating agreement form

A form for updating agreement accompanies all software. Updating for a period of one year is included in the purchase of the program. The agreement shall be signed by the customer and returned to the supplier.

To obtain a system that can utilize improvements and additions to the software, all system software should be continuously updated. We will stress the importance of always having updated software. Besides, there is always the risk that new versions of specific software will not function satisfactorily with older versions of other software.

The updating agreement states that:

* Software is updated about twice a year.

* Updating of the documentation is done at the same time as the software.

1.5 Installation

After the unpacking and placing of the system on a suitable place, the installation, the first upstart, and the system initiation should be done according to the brief list below. More detailed information is given in the following chapters and in the **SA** manual.

Always read the Release Notices for the software. These inform about changes and possible deviations from the description in this manual.

Chapter 2 in this manual should be read before any system connections are made. Before turning on the power, also study the beginning of chapter 3.

1. Before connecting the power cable to the system, check that the main switch on the back is turned off.

2. The key shall be inserted and turned to position STANDBY on the front panel. During the transport, the key is fixed on the back of the computer. The protective cardboard disc in the diskette unit should have been removed when the system was unpacked.

3. With the enclosed cable a terminal is connected as main console in the 15-pole pin plug socket at the lower left backside on the main device (figure in chapter 2). Make sure the correct cable is used (chapter 6).

The terminal shall be locally adjusted to 9600 Baud, 7 data bits, even parity, and 1 stop bit. Normally XON/XOFF is used at communication. The system ignores incoming parity but generates a parity bit at output to the terminal. Therefore also 8-bits and no parity may be selected on the terminal, if it ignores the eighth data bit. Other terminals to be connected, however, may have other communication parameters. See chapter 4.8 for more details.

4. A printer may be connected to the socket to the right of the main console socket. If the used printer cannot be adjusted in the same way as the terminal above, certain system files must be changed before the printer can be used. See below.

5. The system may now be started according to chapter 3. The main power shall be turned ON and the key on the front panel is turned to AUTO. It takes a while for the system to be loaded from the internal disc unit (the winchester unit). During the loading, text is displayed on the main console according to chapter 3.

6. On the main console the text 'Console login:' is displayed, telling that the system is ready and the operator may login by entering 'root' followed by pressing RETURN once. After login a character (#) is displayed on the screen, showing that normal commands may be entered. In chapter 5 a brief introduction to the file system can be found.

7. A normal system shutdown is performed by first turning the key to the position MANUAL and then giving the command '**/etc/shutdown -k**' from the main console terminal. Then it takes about 5 minutes before the system is completely turned off (longer if other users are logged in). The system is completely halted when the text **'OS INFO: System halted'** is displayed on the main console screen. If the system shall be entirely closed, the power can now be turned off by the main switch. However, if the system environment permits so, the system is normally never closed down, except for service or special system maintenance. See chapter 3 for details and other ways of halting the system.

8. Now the system shall be initiated according to the description in chapter 4 before any application programs are run. Above all, users shall be defined **(mkuser)** and passwords set **(passwd)**, also for the user 'root'.

9. If other modifications than the above are required, an unexperienced operator should practise the use of the command **siv**. An example is displayed in chapter 4, and in Appendix A is found a complete and detailed description of the **siv** command, with a brief help list at the end. Before starting the practice the operator should log out with the CTRL-D command and log in again as a normal user with another name than root. Normal users have no authorization to modify certain sensitive system files, which makes it easier to prevent misstakes. Only the 'root' user has super-user privileges and may change the system files. Note that **siv** always can be cancelled with CTRL-C without updating the edited text in the file.

Note also that the shell variable TERM must be defined and adjusted to the terminal in use. Otherwise **siv** does not function. The possible variable names can be listed from the file /etc/termcap by the following command, giving one line for each terminal type.

```
fgrep '|'  /etc/termcap
```

The lines have several text strings enclosed by the character '|'. The second text string 'twist' in the example below is the value of TERM for the described terminal. Example of a line listed with the above command:

```
dl|twist|vt-100|pt100|pt-100|dec vt100:|
```

The most common terminal types are adm3a, vt100, twist (24 row Facit Twist), twi72 (72 row Facit Twist), vt220 and vt240.

Assume that a Facit Twist is used as terminal in a 24-row mode. The following commands are then given to start **siv** to change or create a file we call testfile.

```
TERM=twist      (Ex: 24-row Facit Twist)
export TERM
siv testfile
```

10. Common modifications that usually have to be done:

- Enter the correct definition of TERM in the file /.profile by using **siv**. TERM is then defined automatically when the user logs in as root. TERM will be automatically defined in the .profile in each user's home directory when the user is created. See SA.

- Possibly the file /etc/timezone needs modifying with **siv**, for summer or winter time. The file shall normally contain one line MET-1 for winter time and MET-1MDT for summer time. (See SA)

- If several terminals shall be used, the files /etc/inittab and /etc/-gettydefs may need modification. Possibly new units need to be created in the directory /dev with the command **/etc/mknod**. See chapter 4.6 and **SA**.

- If the used printer can not handle 9600 Baud, 7 data bits, even parity (or 8 data bits, no parity), the file /etc/rc can be modified for automatic start of the process /etc/setspeed. This process can set up the printer parameters automatically at every time the system is started. See chapter 4.7.

- The system is delivered with two printer spooler systems. Standard is **lp,** but an alternative system **(lpr)** is also included. If the **lp** system shall be used, a standard routine, main, is avaliable. This shall be activated with the command '**/usr/lib/accept main**' before the **lp** command can be used. If the **lpr** system shall be used, the start sequence in the file /etc/rc must be modified, and the command **/bin/print** shall be modified in order to refer to **lpr** instead of **lp**. See chapter 4.7 and SA.

- The D-NIX system configuration parameters may need adjustment. To get the maximum system performance, consult your dealer for making necessary parameter changes. See **SA**.

11. It may be necessary to define different user groups in order to have common access to certain files. See Appendix B, which gives some hints on the system setup.

12. When the system is ready and initiated, a total backup should be done on tape casette or on diskettes and stored in a locked safe, if possible in another building. See chapter 5. Backup should also be done regularly of all parts of the file system that are used daily.

13. If the development software package or other application programs are included at delivery, these are normally delivered on separate diskettes and must be loaded and initiated by the operator according to the description (Release Notice) for each program.

14. See **DM** for installation of the Menu system on the system.

2. Assembling the system

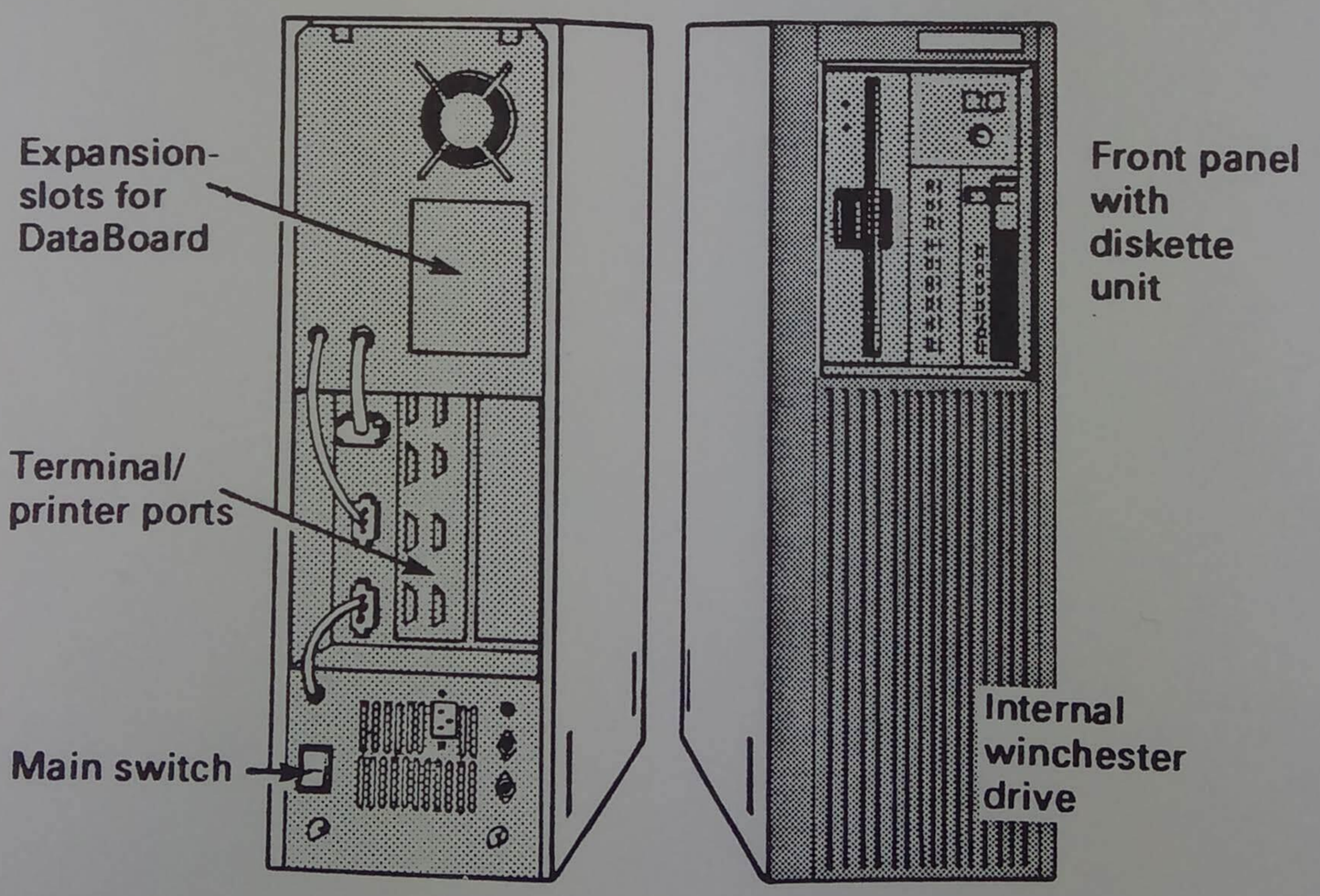
2. Assembling the system

The system is built in a vertical cabinette. On the front there is a control panel with LEDs and a keyswitch, one 8 inch diskette unit and, optionally, a cassette streamer.

Inside the cabinette is a memory of winchester type and also the computer module with a power supply. There is space for expansion of the memory and for installing different options.

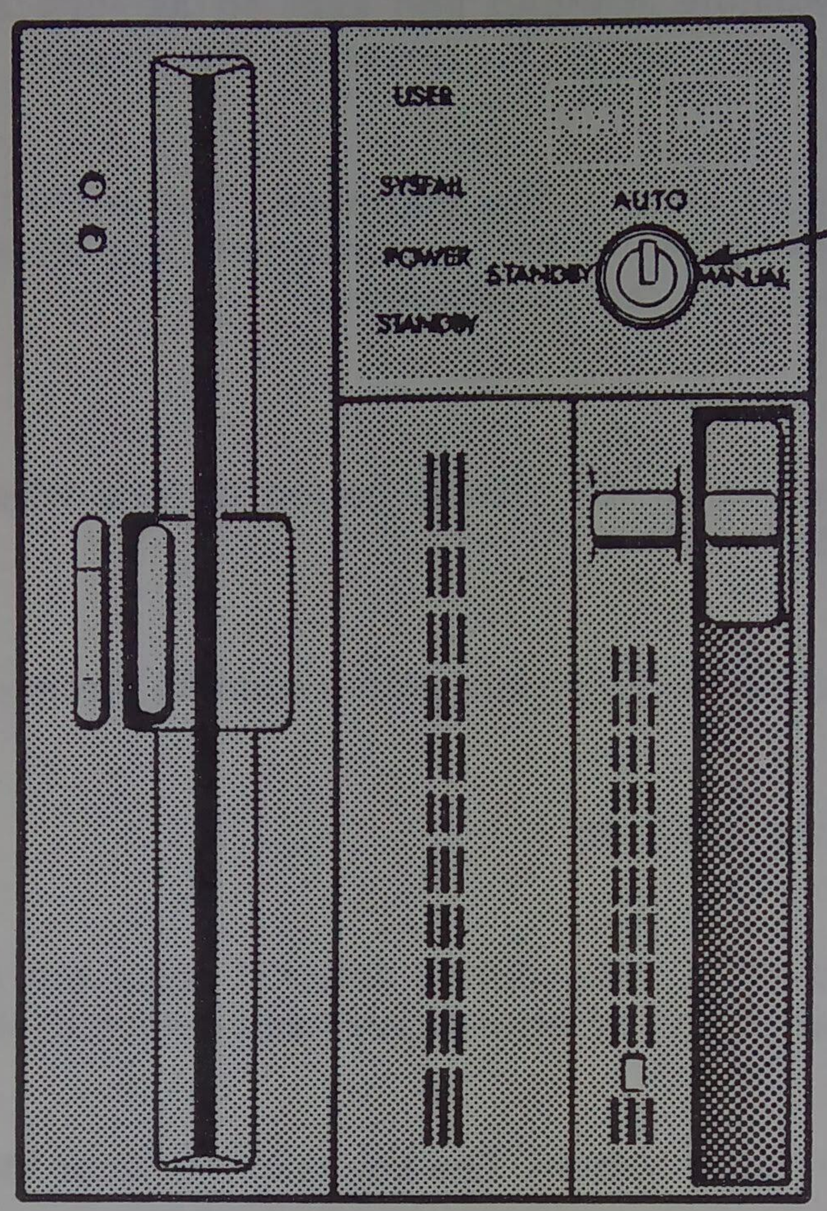
All connectors for power, terminals, printers, and all other units, are located on the back panel.

The system shall first be moved to its permanent location before any connections are made. Space is required behind the computer to reach the backpanel with its connectors. When placing the system, make sure that the environmental requirements mentioned in chapter 1 are satisfied.



2.1 Operating functions

2.1.1 Functions on the front panel



Key switch

AUTO
Auto-mode. The system is automatically started when the power is turned on with the main switch or when the key is turned from STANDBY to AUTO, with the main switch on. Also automatic start 10 seconds after a temporary close down. The push buttons INIT and NMI are not active. At autostart the entire start-up procedure is performed and login may be done from all terminals.

MANUAL
Manual mode. System start-up can be done step by step from a selected unit. The push buttons INIT and NMI can be used.

STANDBY
With the key in the left position (STANDBY), the system is normally shut off. If the main switch is on, the system is in standby mode and may be started by turning the key to AUTO or MANUAL.

Before turning the key to STANDBY it is essential to close down the system by '**/etc/shutdown -k**' and wait for the message '**System halted**' on the main console. Before close down, turn the key to MANUAL to prevent autostart.

Push buttons

INIT
Reset of the system. (Can only be used with the key in MANUAL). The system is immediately stopped and kept so as long as the button is pressed. When the button is released the system restarts again.

NMI
System interrupt is sent to the computer. The button is only sensed with the key in MANUAL. It has no function in the standard system but can be used during program development.

Lamps

All lamps are turned on for a short period of time when the system is started, lamp-test.

USER
User-mode. It flashes when programs are running (green).

CPU2
Reserved for future use (red).

SYSFAIL
System fault. Is lit at start during internal tests but fades out when the system is loaded and no errors have been detected (red).

POWER
The power is on in the system. The key is in AUTO or MANUAL (yellow).

STANDBY
Standby. The system power is off. Is lit when the main switch is on, but the key switch is off (STANDBY). It is turned off a few seconds after system start (green).

Functions on the 8 inch diskette unit

Diskette type
8 inch standard diskette. Diskettes with 1 Mbyte are normally used.

LED, red
Is turned on when the computer writes or reads on the diskette.

Lock
Is pushed over the diskette when a diskette has been inserted. When the diskette shall be removed, the push button to the left on the diskette unit is pushed. **NOTE!** Never try to remove a diskette when the red LED is turned on, except in certain cases when the system is used in stand-alone mode (boot level 1) after a manual start-up.

The diskette is inserted with the lable turned to the right/outwards.

Functions cassette straemer (OPTION)

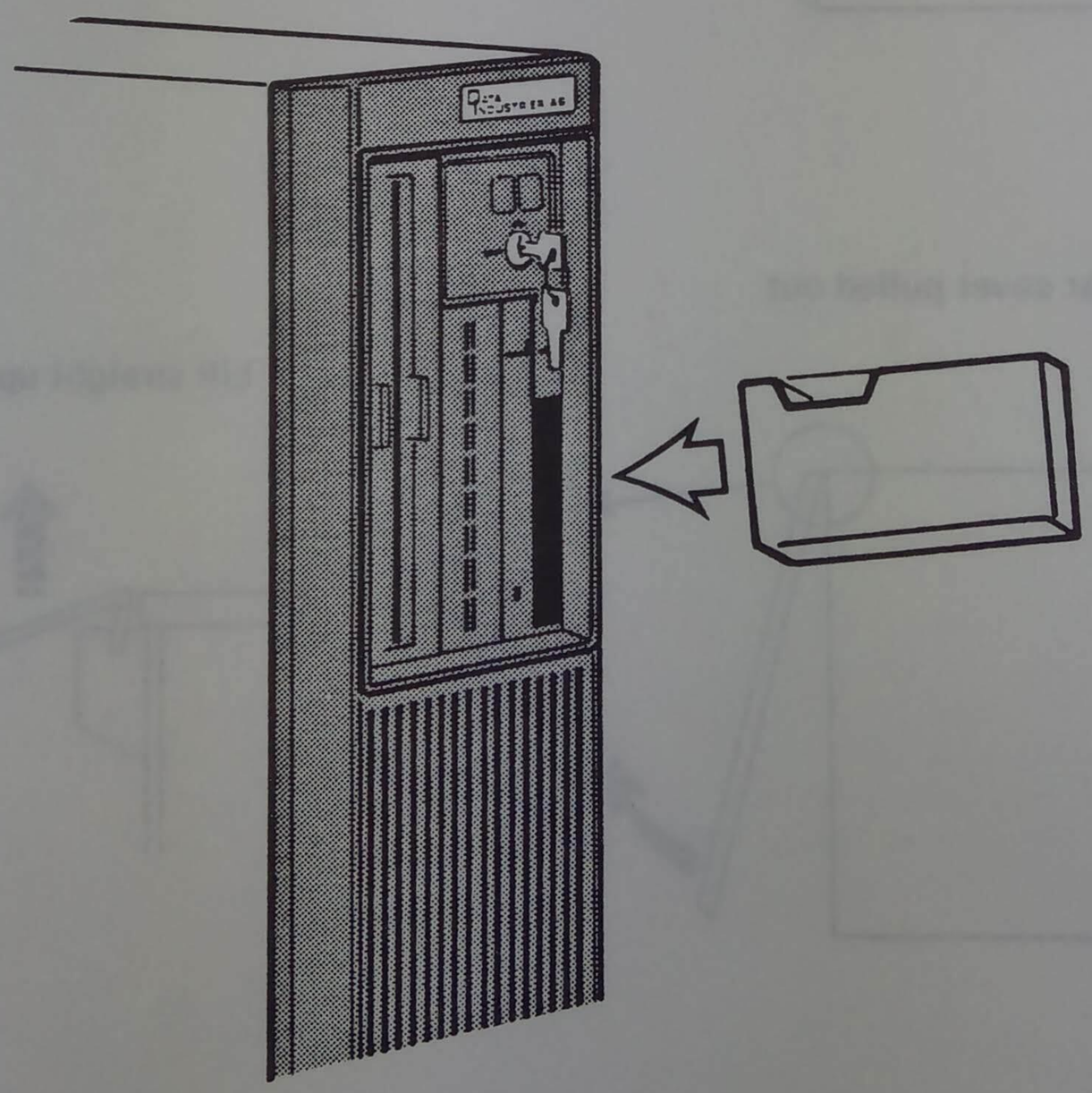
A cassette streamer is not standard, but is a usual option, mounted in the front panel.

Cassette type
1/4 inch tape cassette (e.g. of 3M type) for at least 10 000 BPI. Storage is with 8000 BPI (bits/inch) in 9 tracks according to the QIC-24 standard.

LED (red)
Is turned on when the computer writes or reads on the cassette.

Lock
A sliding lock that locks the cassette in the lower position. It shall be slided up to release the cassette, and pushes the cassette out a little. **NOTE!** Never try to take out the cassette when the red LED is turned on.

The cassette is slided into the opening with the shutter for the recording head facing up/inwards.



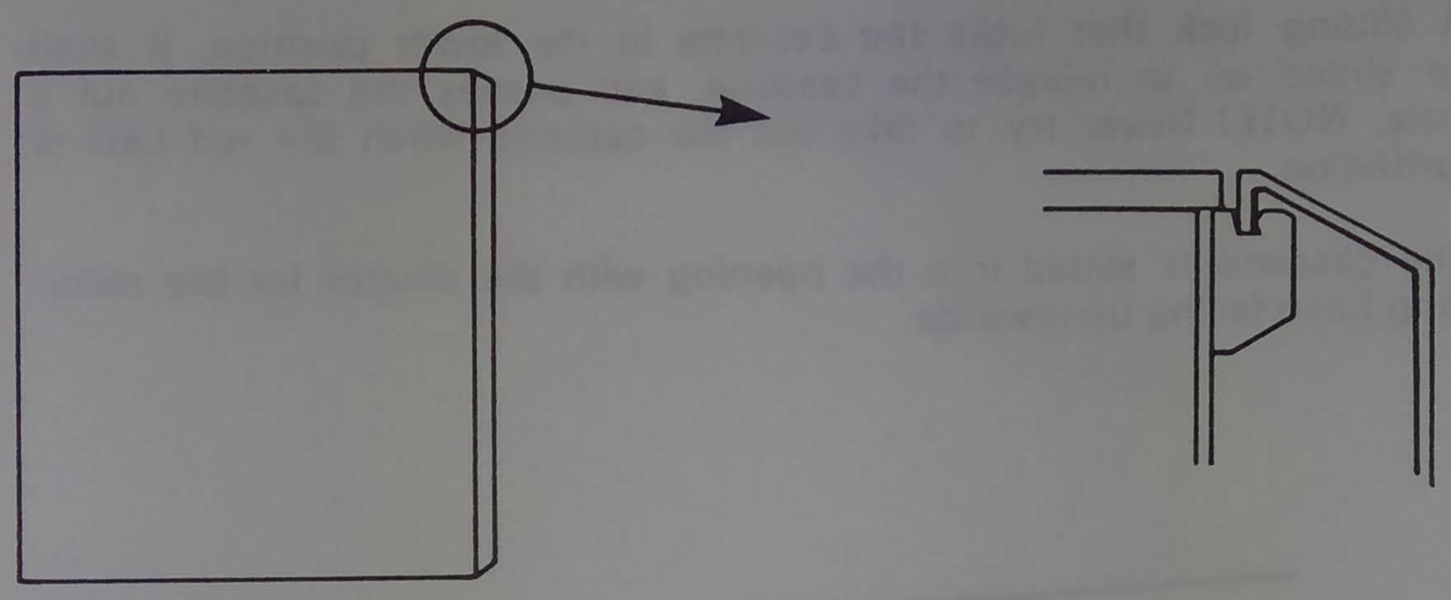
2.1.2 Connectors and operating functions on the backpanel

All connectors are behind the cover on the back. To reach them, the rear cover must be removed. The cover is attached on the top end to a groove and is secured with a snap lock at the bottom edge.

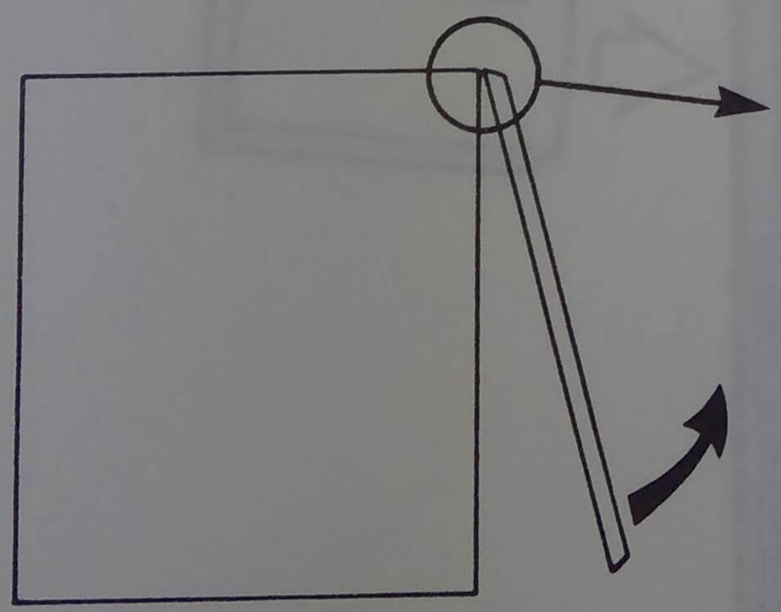
* Lift off the cover by first pulling at the lower edge, thereafter lifting the cover straight up.

The system, seen from the side, where the rear cover shall be removed.

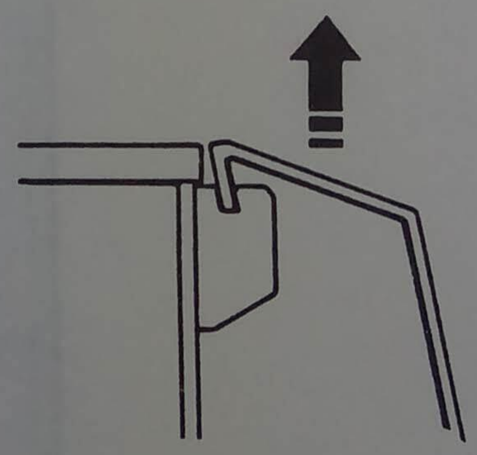
The rear cover attached in the normal position



The rear cover pulled out

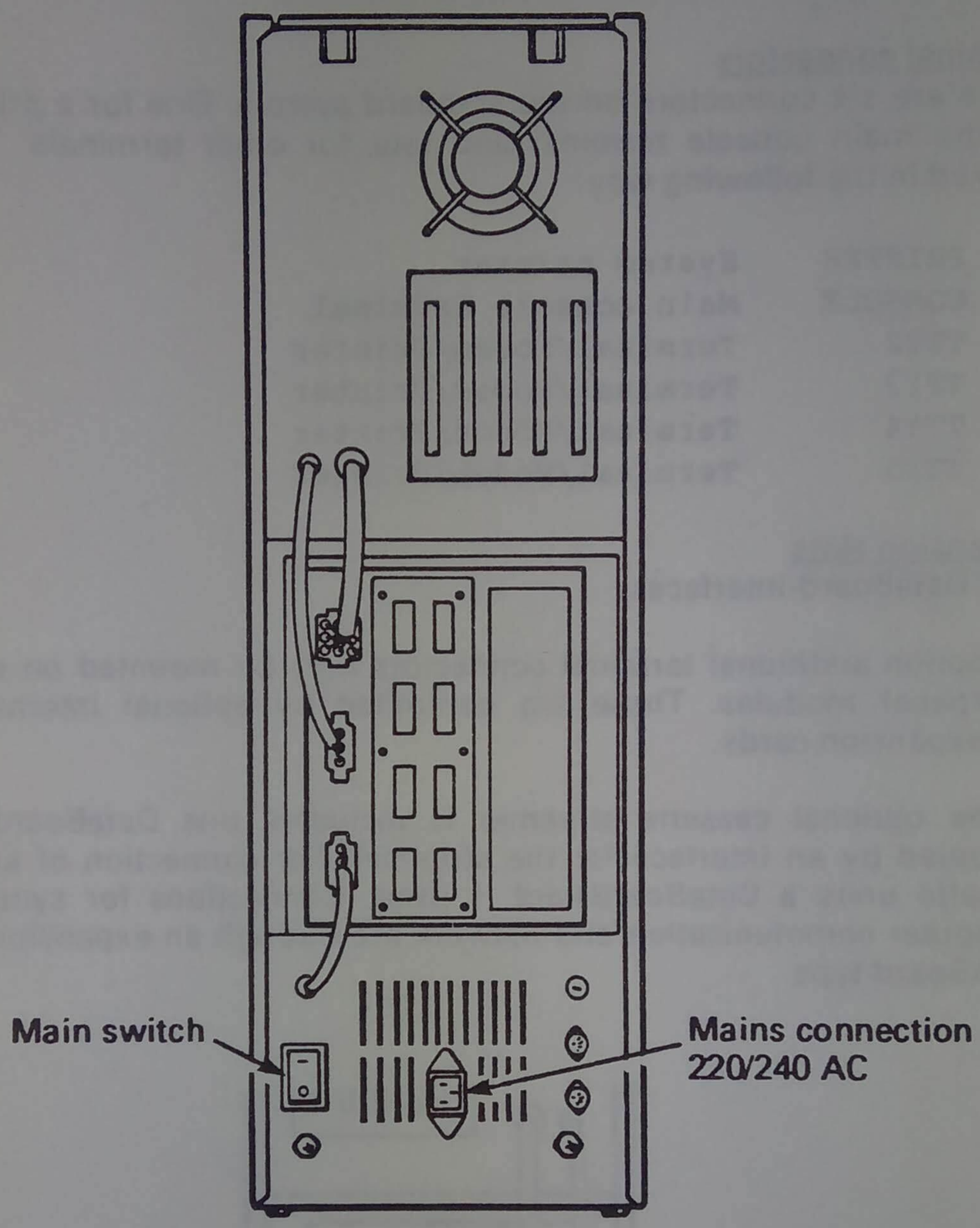


Lift straight up



Behind the front a key is fastened with tape. The key is for the key switch on the system front. Take care of this key, it is with the key the system is started.

The rear panel on the system looks like this:



* Check that the main power switch is in the OFF position.

* Check that the key switch on the front is in the position STANDBY.

* Connect the power cable.

An explanation to the different connectors is given on the next page.

Main power switch
Off/On (MAIN, 220/240V)

Remote power control
Through the two connectors on the lower part of the back panel, several systems can be interconnected to enable start-up and close down with one common switch. (REM OUT, REM IN).

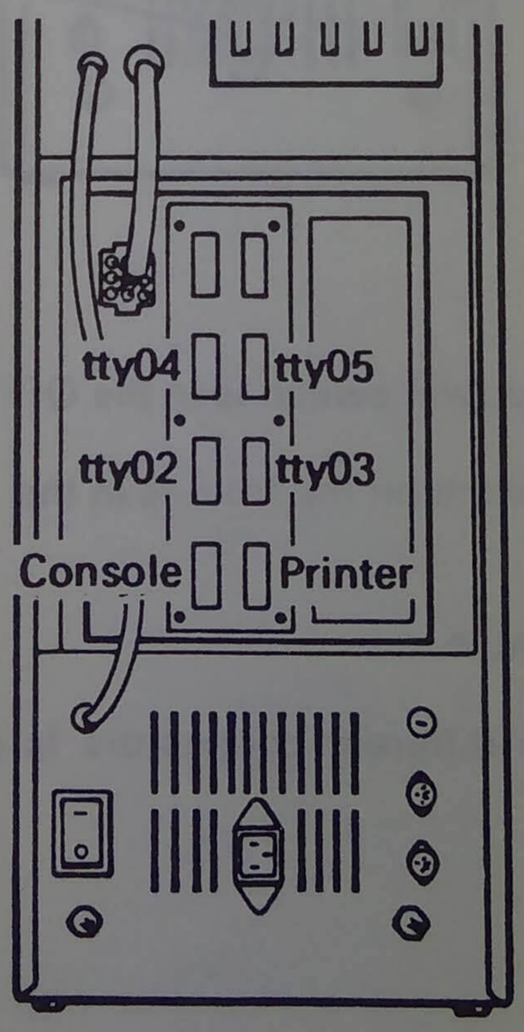
Terminal connectors
There are six connectors on the standard system. One for a printer, one for the main console terminal and four for other terminals. They are marked in the following way:

PRINTER	System printer
CONSOLE	Main console terminal
TTY2	Terminal/Modem/Printer
TTY3	Terminal/Modem/Printer
TTY4	Terminal/Modem/Printer
TTY5	Terminal/Modem/Printer

Expansion slots
Five DataBoard-interfaces.

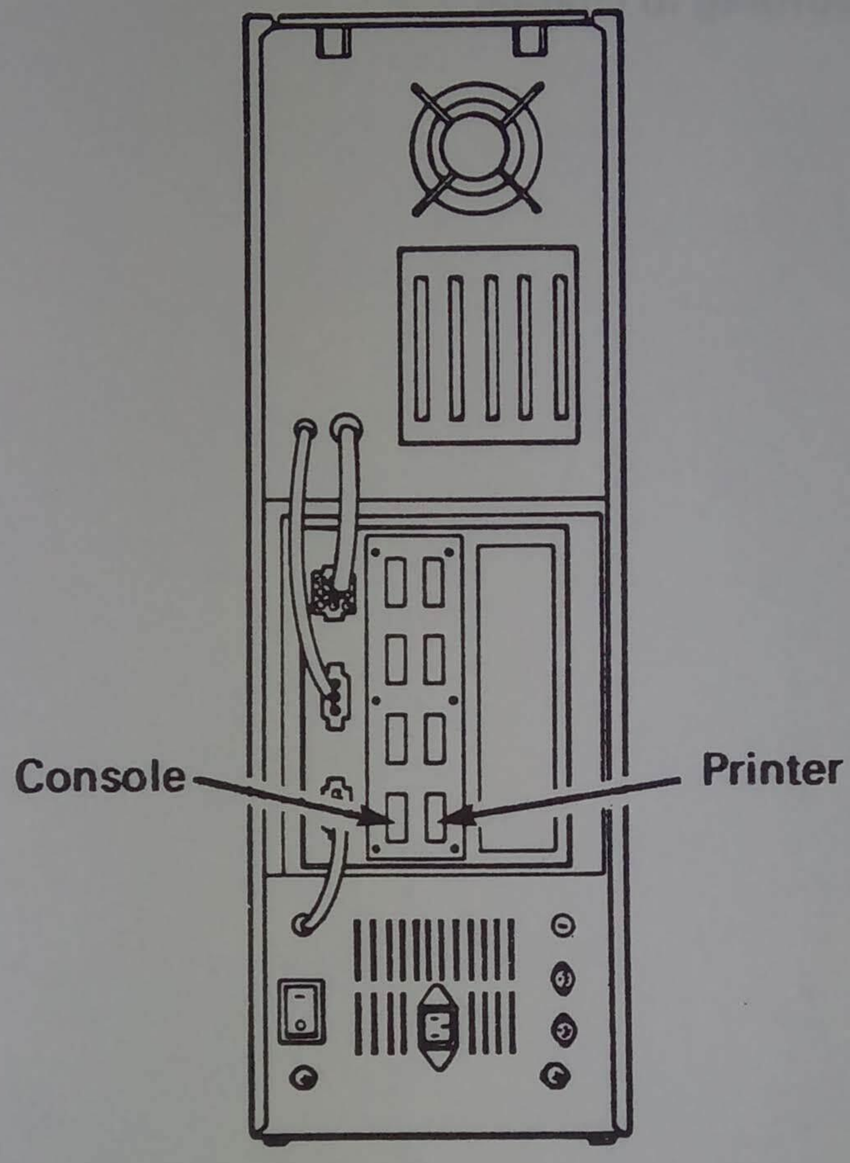
As option additional terminal connectors may be mounted on extended backpanel modules. These are controlled by optional internal terminal expansion cards.

If the optional cassette streamer is included, one DataBoard slot is occupied by an interface for the streamer. For connection of additional diskette units a DataBoard-card is used. Connections for synchronous computer communication and network are through an expansion card of DataBoard-type.



2.2 Connection of terminals and printers

In the basic system there is a total of six serial ports for terminals, printers or modem. The main console terminal must be connected to one port and the system printer to another. As terminal, several different types of terminals can be used. In chapter 6 you find an overview of which terminals the system accepts and also what cables to be used. Several different printers can be connected to the system. The cables to use are listed in the same chapter.



* Check that the power switch on the backpanel is OFF.

* Connect the main console to CONSOLE.

* Connect the printer to PRINTER.

* All other terminals, printers or modems are connected to the other available terminal connectors. **NOTE!** Make sure the correct cable is used (Chapter 6).

* Set the main power switch ON.

Note! The cables should not be connected with power on.

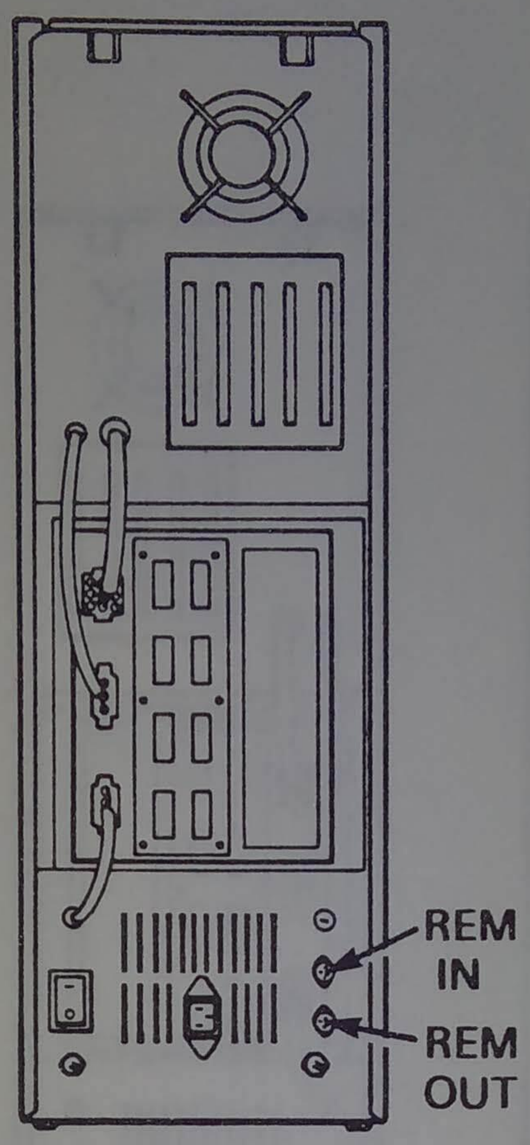
The system is in STANDBY mode, i.e. the start-up logic is the only part having power. On the front panel this is indicated by the green lamp STANDBY. The system is now connected and ready.

Note! If the system is not at a normal temperature after a transport, it shall not be started until the temperature has reached a normal room temperature.

* Re-mount the rear cover plate on its position.

The connected cables are pulled out below the cover on the back. The system can now be placed on its normal site but note the environment requirements according to chapter 1.2.

2.3 Remote power control



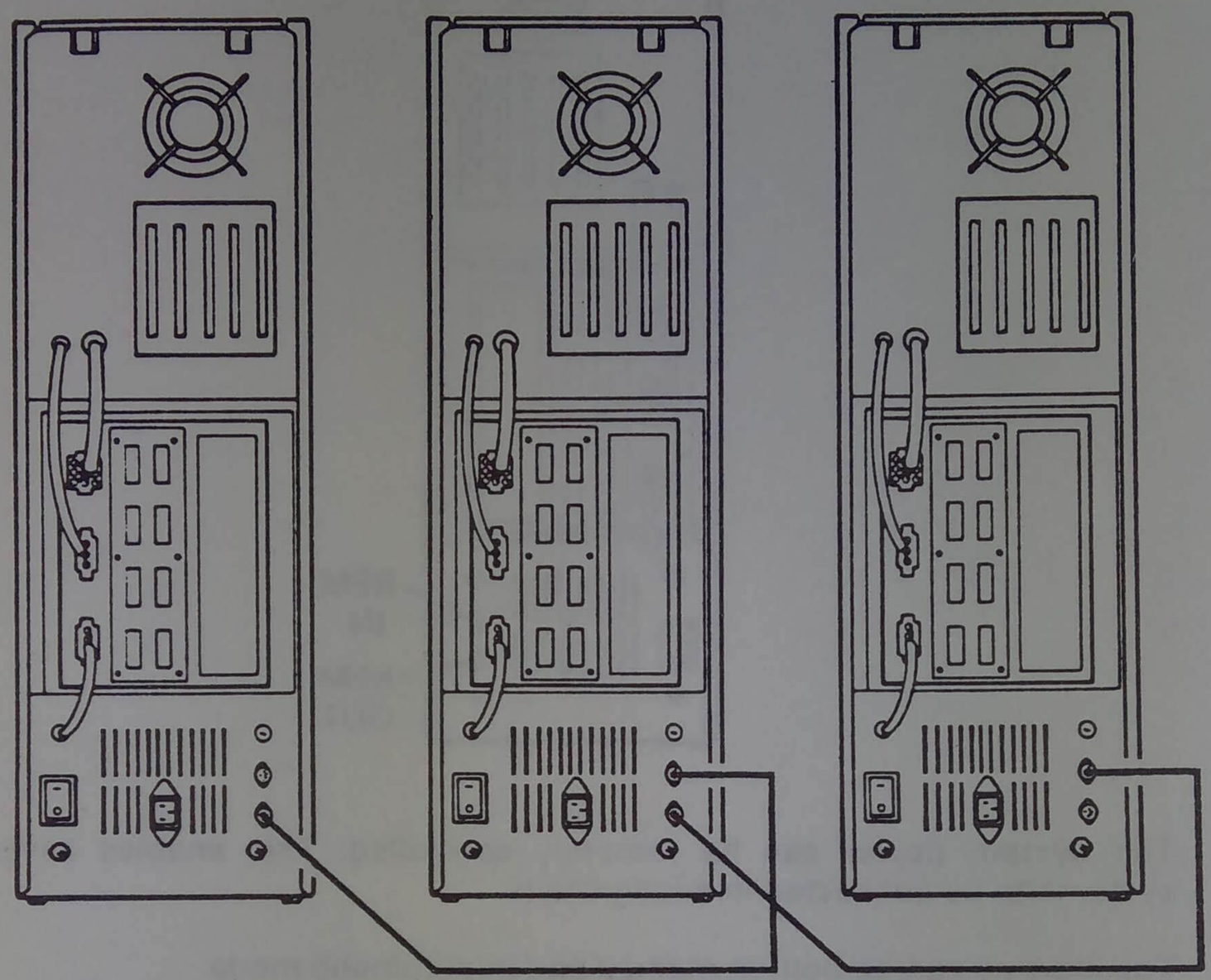
The system power can be remotely controlled. This enables several systems to be connected in a long chain.

The start-up can be both in manual and in automatic mode.

One system is the main system. It is from this system the power is controlled.

The systems are interconnected with a three-wired cable.

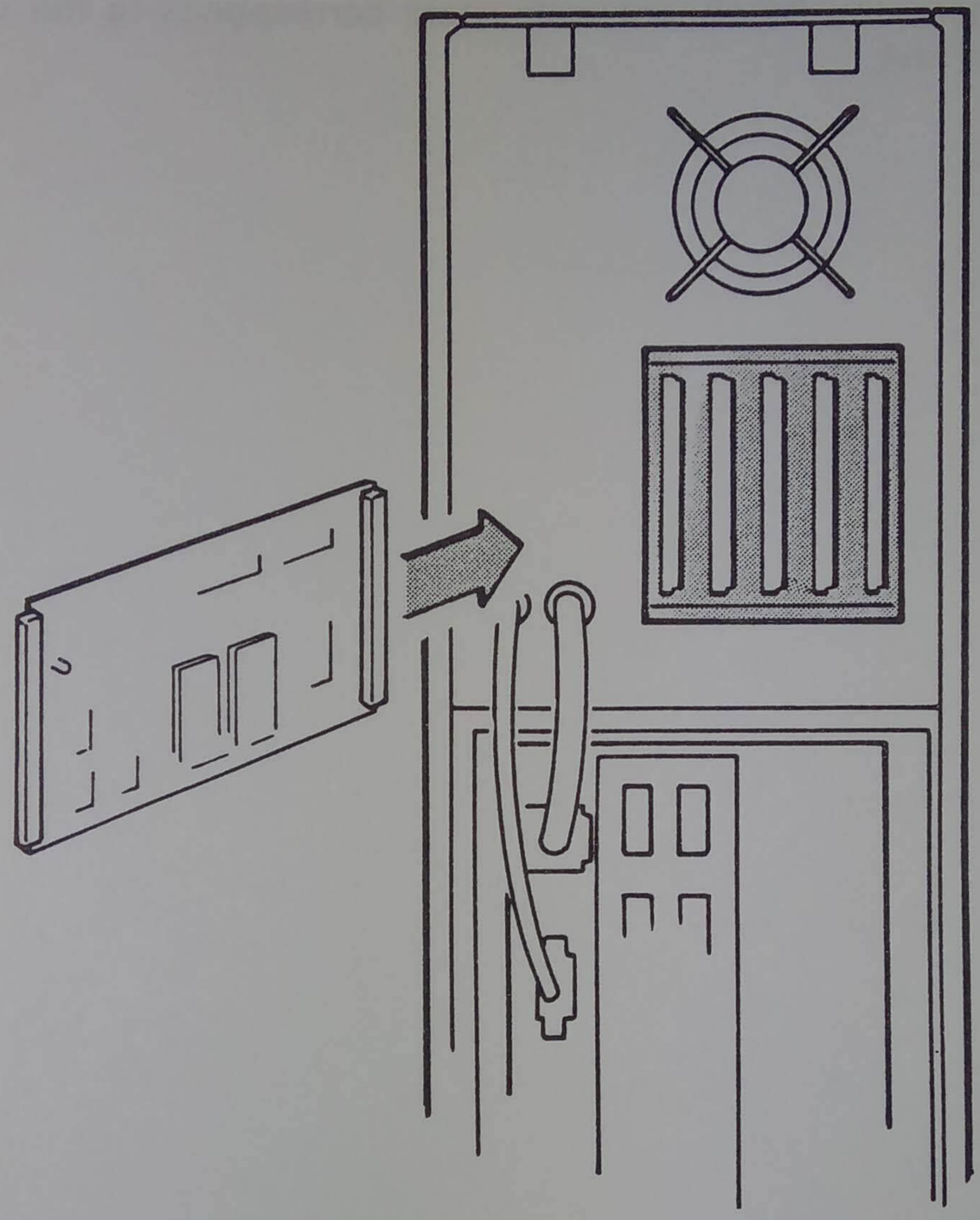
On each system there are two identical connectors (REM OUT, REM IN). The master end of the cable shall be connected to REM OUT on the main system and the slave end is connected to the next system (REM IN). If several systems shall be controlled, the next system (REM IN) is connected to REM OUT on the previous system in a chain.



When all system are interconnected, all systems are controlled from the front panel key on the main system. The next chapter described how a system start is done.

2.4 Connections to the expansion rack

Up to five DataBoard modules can be installed in the expansion rack.



Installation of DataBoard cards are done in the following way:

Before the installation a number of jumpers shall normally be closed on the DataBoard card according to bypacked instructions. All DataBoard cards shall have a channel select code on jumpers. See **SA**.

Check that the system is closed down and the main power switch in the position OFF.

Only the first position to the right, seen from the back, has extra signals for an expansion module to an external rack.

The DataBoard cards are slided into the rack with the component side to the right and the LED outwards.

External cables are then connected according the descriptions for each DataBoard card, before the power is turned on.

The pin layout in the DataBoard connectors corresponds to the I/O-bus in a DataBoard rack.

3. Start of the system

3.1 Start of the system

3.2 Close down of the system

3.1 Start of the system

The system is now connected and ready for start. If more units is delivered with the system, these shall be connected according to the instructions in the bypacked manuals and the Release Notices before start.

After transportation, the system must have reached room temperature before it is started.

* Turn the key to the AUTO position.

The system is now in the Auto-start position, i.e. when the power is turned on with the main switch on the back, the system will automatically be started.

The basic operating system is delivered installed on the internal winchester disc unit. This normally makes it possible for the user to start directly with automatic start.

If, for some reason, the system does not have a generated operating system on the disk, the system must be started from a diskette. After such a starting procedure the operating system should be installed on the hard disk.

The following sections describe step by step the first start up, both from hard disk and from diskette.

3.1.1. Normal system start from hard disk

The system can be started in two different modes, automatic mode or manual mode, depending on the key-switch on the front panel. In manual mode the operator controls the start-up procedure step by step by commands, while in automatic mode the system itself performs the entire start-up procedure. **SA** gives a detailed description of what happens within the two modes.

Below only the automatic mode starting process is described as this is the normal way to start the system.

Check that the terminal connected to the system is properly set, i.e baudrate 9600, 7 data bits, even parity, 1 stop bit, and gives the required signal DTR from the terminal. See the terminal cable in chapter 6 and **SA**.

* The system is connected, with the main switch ON, but in standby mode. The key is in the position STANDBY.

* Turn the key to the AUTO position, upwards.

WAIT! The system is now being loaded from the winchester unit.

On the front panel all four lamps are turned on. After a short while SYSFAIL is turned off. Then the lamp DISC will flash while the system is loaded. After the message below, only POWER will be lit.

When the system is loaded and ready, a message is displayed on the main console which may look as follows:

```
OS:INFO:D-NIX X.X Virtual Ver X.X -
        System xxxk User xxxxk Swap x.xM
Diab Data AB  D-NIX 5.X Y.Y Virtual
Wed Oct 5 15:16:45 MET 1986
CRON started
LP scheduler started

Console Login:_
```

If so far everything has worked out correctly, proceed to section 3.1.3 for login to the system.

If any problem arises during the autostart it may be due to some of the following:

* The main console is not correctly connected or initiated. Possibly, the communication parameter for the main console has been changed in the file /etc/gettydefs. In this case, the start can always be done n imanual mode up to the single user mode with the standard parameers tabove, which in turn makes it possible to modify the file /etc/gettydefs. See the chapters 4.8 and 6 and and the manual **SA**.

* The system has not been halted properly at an earlier working session. At a new start, the operating system will check the file system and correct eventual errors. Several different messages are then displayed on the main console at the start. See the command **fsck** in **UM** for more information.

* The system programs on the winchester unit are incorrect. Start the system in manual mode (see section 3.1.2).

* If the lamp SYSFAIL does not go out as expected, it might be due to a hardware error found by the automatic diagnostic program. If so, a service engineer should be consulted.

3.1.2 System start from diskette

The system may also be started from the diskette unit. This is necessary if the winchester unit for some reason is not working. Manual start is described in detail in SA, but below follows a short description of the sequence.

The system shall be totally closed down, the key in position STANDBY but the main switch on. Manual start can also be done directly after a termination, as soon as the message "OS INFO: System halted" has come up on the screen, without awaiting the power to be turned off.

Now press the button marked INIT. This gives a reset of the system. The computer will display the following on the main console screen:

```
M68000 prom
```

To start from diskette the diskette marked **"D-NIX 5.2 DS90-00 / BOOT mount format"** is now inserted in the drive. When this is done, give the following command:

```
sf(0,0)
```

ended with <RETURN>. The stand-alone system is then loaded from the diskette, after which the system asks:

```
System boot
Root device "xx(nn,nn)"?  sf(0,0)
```

where sf(0,0) is the user's answer to the question.

The system gives a prompt ">" and the operator starts D-NIX by writing 'dnix' ended with <RETURN>, and the D-NIX system is loaded and started in single-user mode (boot level 2).

```
>dnix
```

NOTE! When the D-NIX operating system has been started with the diskette as rootdevice, the diskette MUST NOT be taken out of the diskette drive before a correct system termination has been performed. It is not possible to enter the D-NIX multi-user mode (boot level 3) with only the BOOT diskette as root device, as the space on the diskette is limited.

Manual system start can also be done from the internal winchester disc by answering only RETURN instead of the unit name for the diskette. In this case the boot program reads the stand-alone system from the unit given in the system NVRAM, normally the winchester disc. See /etc/-bootpar in **UM** or /sas/bootpar in **SA**.

3.1.3 Login and defining password for root

At the first start up of a system from the internal disc unit, the login procedure is slightly different from the normal login procedure. Normally the system asks for both login name and password, but in this case only the login name is required. This is because every individual user, himself or aided by the system administrator, should be able to define his own password.

To protect the system from unauthorized users, the system administrator already at the first contact with the system should define a password for the user root.

The following is displayed on the screen:

```
OS:INFO:D-NIX X.X Virtual Ver X.X -
        System xxxk  User xxxxk Swap x.xM
Diab Data AB  D-NIX 5.X Y.Y Virtual
Wed Oct 5 15:16:45 MET 1986
CRON started
LP scheduler started

Console login:_
```

* Enter root. Terminate the input by pressing <RETURN> for login to the system.

NOTE! The user name root must be written in lowercase letters.

When the system is ready the character # is displayed on the screen. This character is called prompt and indicates that it is now possible to work with the system, for example to define passwords and new users, to add/delete file systems, and so on.

The prompt # also points out that the user works in super-user mode. In D-NIX, a user may work in different modes and each mode has its own prompt character e.g # for super-user mode.

Before the question 'Console login:', the nodename of the system may be displayed and a system message from the file /etc/issue, if these are defined. After accepted login, before the prompt (#), possible other system messages may be displayed from the file /etc/motd.

NOTE! Be careful when working in super-user mode, as many of the locks on D-NIX commands are disabled in this mode.

It is assumed in the rest of this chapter if not otherwise declared, that the user is working in super-user mode and that the root directory is the current directory, if nothing else is stated.

Defining a password for root

As the first start up is a special one and the system does not ask for a password at login, it is of importance to define a password at the very first contact with the system. A password is defined by using the command **passwd**.

A password must contain at least six characters, of which at least two shall be alphabetic and one shall be a digit or a special character. If more than eight characters are entered only the first eight are significant. The system asks for the password twice to ensure the proper spelling. If the system does not accept the password, this means that it has not been spelled the same way at both entries. Note that upper case and lower case letters are interpreted as different characters by the system. If the first attempt fails just re-use the command **passwd**.

* Enter **passwd**. Press <RETURN>.

* Enter the password. Press <RETURN>. NOTE! Use at least six characters.

* Enter the password again. Press <RETURN>.

The screen may look like this:

```
passwd <RETURN>
passwd: changing password for: root
new password: (enter the password here) <RETURN>
retype new password: (once more) <RETURN>
```

passwd	The command itself.
new password	Enter new password, min six charact. NOTE! The password is never echoed on the screen.
retype new password	Enter the password once more. NOTE! The password is never echoed on the screen.

When the password has been accepted, only persons who know both the login name and the password are able to login as super-user.

3.1.4 Log out - exit from the system

Exit from the system is done, either by pressing CTRL-D or by giving the command **exit** or **exec login** to the system.

```
# _
```

* Press the keys CTRL and D simultaneously.
  (or enter **exit** and press RETURN)
  NOTE! Nothing is written on the screen.

After a short while **login** is displayed as:

```
#
Console login:_
```

The command processor is stopped and 'login:' is written on the screen for login by another user. At exit from the system all open files are closed. If a background process is running, the system does not wait for its termination. If the command 'exec login' is given, a new login will occur, but without redefining the terminal parameters. However, command statements in the file .profile may affect the terminal.

3.2 Close down of the system

It is important that the power-down of the system is carried out in an orderly way. All users must be enabled to log out from the system and all files and file systems must be updated before they are closed.

This is carried out by an operator with super-user privilegies from the main system console according to the following:

* Turn the key to the position MANUAL.

* Give the command '**/etc/shutdown -k**'

* Wait until the text "**OS:INFO: System halted**' is shown on the screen. With the key in the MANUAL position, the text 'M68000 prom' may be shown, but this does not matter. If no users are logged in, all processes are terminated in an orderly way.

* With logged in users, each user gets a message each minute that the system shall be closed down within a certain time. After five minutes, the system is closed down, forcing logout for anyone still in the system.

The system is completely closed down when the following text appears on the screen:

```
OS:INFO: System halted
```

* If the key was left in AUTO during the close down, it must now be turned to STANDBY (or MANUAL) within 10 seconds after the text '**System halted**', to prevent a new automatic start-up.

* Turn the key to the STANDBY position.

* Wait until the system is entirely closed, after which the main power switch on the back can be turned OFF.

When **shutdown** has been started, it must be permitted to run until completion. It must NOT be interrupted by BREAK or DEL.

Closing down the system from the bootlevel 0 (boot) and the bootlevel 1 (stand-alone-system) can be achieved at once by turning the key to STANDBY, provided that no program is writing or reading the disk memory. In these cases also a system restart may be performed by pushing the INIT button (with the key in MANUAL). The manual start through the boot levels 0, 1, 2, and 3 is described in **SA** where 2 and 3 correspond to single-user and multi-user levels of the operating system.

There are three normal ways of changing the system configuration:

1. Complete halting by the command '**/etc/shutdown -k**' as above, after which also the power is turned off.

2. Restart of the entire system with the command '**/etc/shutdown -k**', followed by automatic or manual start-up.

3. Going down from the multi-user mode to the single-user mode, which is done with the command '**/etc/shutdown**'.

shutdown is a system command. Its primary function is to terminate all running processes in an orderly way. Normally all processes are terminated and the system goes down to single-user level without a total termination. However, if the option '-k' is given to the command the system will terminate completely. A termination with shutdown normally takes less than 5 minutes if no users are logged in.

shutdown goes through the following steps: First, all logged in users are requested to log out by a message on their terminals. They have five minutes for doing it themselves or else they will be logged out automatically. All mounted file systems are updated during shutdown. This must be done before the next loading of the system or else the system will not be intact.

As parameter to /etc/shutdown may be given a delay time in minutes, to be used instead of the usual five minutes for user log out. Further a message for the user may be given and displayed together with the announcement from **shutdown** to terminate.

shutdown requires super-user privileges and that the operator is logged in as 'root' on the main console.

At a total termination with '**/etc/shutdown -k**' the operator must wait until the system has written the following on the console terminal before the key switch is turned to the position STANDBY and/or the power is turned off.

```
OS:INFO:System halted
```

4. Configuration of the system

4.1 Create and modify system files with siv

4.2 Add user - mkuser

4.3 Add/change password - passwd

4.4 Which users are defined in the system

4.5 Remove user - rmuser

4.6 Add terminal/modem connections

4.7 Printout on printer

4.8 Parameters for terminals, modems and printers

4.9 Initiating a new file system on the internal disc

4. Configuration of the system

In this section a brief information is given on those files and commands that may be needed when adjusting the system to the needs of the user. The commands are explained from the user's point of view. A summary is given in a subsection, telling what factors to consider when connecting a new terminal or printer.

All examples assumes that the user (the system administrator) is in the root directory and logged in as super-user (root), unless otherwise stated.

NOTE! Before you start to modify sensitive system files you should learn to manage the process as follows:

* Improve thoroughly your knowledge of the '**siv**' command and how to use it. Also make sure that the shell parameter TERM is set to the terminal in use.

* Make copies of the system files and modify these copies. After checking that it is properly modified, the original file should be renamed as a backup and the new file be given the name of the original. When copying '**cp**' is used as shown below. With **cp**, the file owner and access permissions are reset to the same as for the original file. The commands **chown** or **chmod** may otherwise be needed to correct the owner and access permissions. For more information about commands, see **UM**.

Example: Modification of /etc/inittab is done as follows:

```
cp /etc/inittab /etc/inittab.new     Copies to a work file
cp /etc/inittab /etc/inittab.old     Backup copy of the
                                     original.
```

Modify /etc/inittab.new using **siv** and exit from **siv** with CTRL-X CTRL-F to save the changes.

```
siv /etc/inittab.new
```

```
cat /etc/inittab.new                 Verifying of changes
cp  /etc/inittab.new /etc/inittab    Move the changed file
                                     to the original file.
```

4.1 Create and modify system files with siv

With the editor **siv** the contents of a file might be modified if the user has read- and write privileges to the file or is a super-user.

The following example describes how the file /etc/motd (message of today) is modified. This file is presented to everyone logging into the system. The editor **siv** is described in appendix A.

Define the parameter TERM if not already defined for the terminal in use. Enter the following for 24-row Facit Twist. No space may be used around the equal-sign.

```
TERM=twist
export  TERM
```

* Enter **siv** /etc/motd. Press <RETURN>.

```
siv /etc/motd  <RETURN>
```

```
siv             The editor
/etc/motd       Filename
```

On the screen the contents of the file /etc/motd will now be displayed. For example:

```
        Welcome to D-NIX

    86-03-10
    Today we have got the new laser printer in function.
    It may be used as printer lp2.

    Comments may be sent by mail to sysop.
    End.
```

We now want to change the name of the printer to laser. Move the cursor to the row to be changed and to the first character in the word lp2.

* Press CTRL and D simultaneously as many times as needed to get the whole word erased.

* Enter the new word laser.

* Press CTRL-X and CTRL-F. The file will be saved and the editor is terminated. The prompt # is displayed.

The file is now modified, which may be checked by the command **cat**. For information about the commands in **siv** see Appendix A.

To terminate the editor **siv** without changing the file press CTRL-C and answer yes.

4.2 Add a new user - mkuser

At the first start-up of the system, only one user is defined in the system, the super-user root. However, everyone to be a user of the system has to be defined. This is done with the command **mkuser**. This command is built up as an interactive program where a number of questions have to be answered.

```
/etc/mkuser <RETURN>
```

See appendix B in this manual for practical hints about organizing users within a company. For removing a user use the command /etc/rmuser.

The dialogue of mkuser:

Username:
The name of the new user to be used as login name. If the name already exists the question is repeated. Avoid using only upper case letters in user names. Also avoid the characters '_','$','&', '/','#' and '@'. These combinations are specially treated in the start routine for login (/etc/getty).

User-id:
The numeric ID of the new user. This must be an integer greater than 1. For users in general a number greater than 100 is recommended. If the given id is occupied the question is repeated. <RETURN> only, gives the first free id number.

Password:
Define the password the user wants to use at login.

Group-name:
The name of the group which the user shall belong to. It is possible to divide users into one or more groups and for each user define which group he belongs to. A password kan be defined for each group. If a user shall not belong to a special group, only enter <RETURN>, which makes the groupname to be defined as 'other'.

Group-id:
The number identification of the group, an integer greater than 1. The system displays the group-id if an already existing groupname has been entered. If the number is occupied the question is repeated. <RETURN> only, gives the first free number.

Extra user description:
Arbitrary string. Colon (:) may not be included.

Login-directory:
Home directory of the user. I.e. the directory the user will come to after login. If it already exists, the answer must be verified and no own .profile may be created. If only <RETURN> is given, the start directory will be /usr/username, where username is the same as above.

Start-program:
Defines what start program to be used. If nothing else is defined, the standard start-program, /bin/sh is used. If the given program does not exist, the answer must be verified.

Terminal-type (vt100):
The type of terminal the user is expected to work with. The commands for defining the shell variable TERM to this type will automatically be inserted in the file .profile created in the home directory. The default type is shown within parenthesis (here vt100). If this terminal is used, press only <RETURN>.

When all questions have been answered the system will define the user, which means that the files /etc/passwd and /etc/group are updated. In addition, a home directory is created as well as a start-up command file .profile, which will be put in the user's home directory. The start-up file will be executed whenever the user logs in. The file .profile contains information of which terminal type is used (TERM) and may also hold function codes for some of the keys. The file /etc/passwd contains the following information about each individual user: login name, ciphered passwd, user identity, group identity, comment field, home directory and eventual start directory. Privileges and time limits for modifying of password can be added manually later on. In the file /etc/group the following information about all groups is saved: group name, password, group identification and users belonging to a certain group. For further details, see **SA**.

Example: mkuser

```
/etc/mkuser <RETURN>
User name: oskar <RETURN>
User-id: 114 <RETURN>
Password: (password) <RETURN>
Repeat password:(password) <RETURN>
Group-name: other          <or just RETURN>
Group-id: 1
Extra user description:Ossian Anderson <RETURN>
Start-directory: usr/osand <RETURN>
Start-program: <start program or RETURN>
Terminal-type (vt100):  <if not vt100 enter terminal-
                          type otherwise RETURN>
```

/etc	The command is found in this directory
/mkuser	The command itself.

The rest of the text is explained above.

The user is now defined in the system.

4.3 Adding/Modifying a password - passwd

Every user has a password attached to his home directory in order to protect it from unauthorized use. The system is delivered with only one user defined, the super-user, and this user has no password defined. Therefore the system administrator should already at the first contact with the system define a password. This is achieved by the command **passwd**.

When a new user is defined the system administrator has the possibility to define a password as well. However, the user himself should have this possibility if it is evident that other users have gained knowledge of the password. Modifying of the password is also done by the command **passwd**.

A password must contain at least six characters, of which at least two shall be alphabetic and at least one shall be a digit or a special character. The system asks for the password twice to ensure the proper spelling. If the system does not accept the password, this means that it has not been spelled the same way at both entries. Try again! Note that upper case and lower case letters are interpreted as different characters by the system.

Adding/Modifying a password

```
passwd <RETURN>
passwd: changing password for: kalle
old password:
new password:
retype new password:
```

NOTE! The password is not echoed on the screen.

```
passwd                  The command itself.
passwd: changing ..... Tells for which user, the pass-
                        word is changed.
old password:           Enter the old password.
new password:           Here the new password should be
                        entered.
retype new password:    Enter once more.
```

A user can be forced to define a new password after an ultimate time in weeks and can also be kept from changing his password. This is controlled by codes in the field with the cihpered password in /etc/passwd and can be defined only by the system administrator by direct editing of the file. See SA and UM for details.

For users which shall be able to login to the system from an external line through a modem, there is a possibility to define a second "dial-up password" for increased security. See the **login** command in **UM**.

4.4 Which users are defined in the system

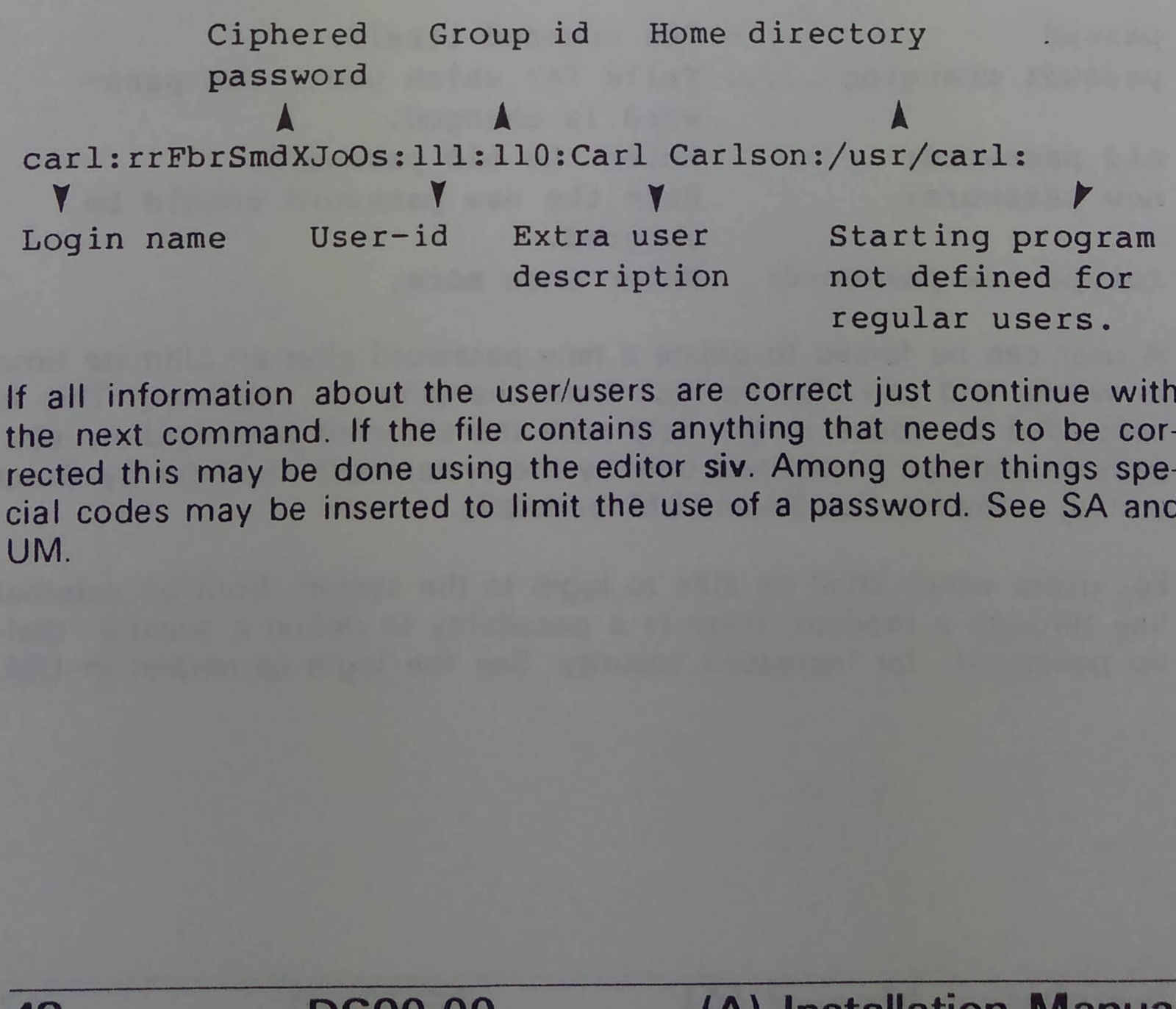
When one or more users have been defined in the system it may be of interest to check that this has been done in a proper way. The easiest way to do this is to print out the information in the /etc/passwd file. Use the command **cat**, which prints out the file contents on the screen. Because of limited line length in the manual long lines are divided in the listing below. On the screen they may look different.

* Enter **cat/etc/passwd**. Press <RETURN>

The screen may look like this:

```
root::0:3:0000-Admin(0000):/:
daemon:*noway*:1:12:0000-Admin(0000):/:
bin:*noway*:2:2:0000-Admin(0000):/bin:
sys:*noway*:3:3:0000-Admin(0000):/usr/src:
adm:*noway*:4:4:0000-Admin(0000):/usr/adm:
uucp:*noway*:5:1:0000-uucp(0000):/usr/lib/uucp:
nuucp::6:1:0000-uucp(0000):/usr/spool/uucppublic:
          /usr/lib/uucp/uucico
sync::20:1:0000-Admin(0000):/:/bin/sync
rje:*noway:68:8:0000-rje(0000):/usr/rje:
shqer:*noway:69:8:0000-rje(0000):/usr/rje:
lp:*noway:71:2:0000-lp(0000):/usr/spool/lp:
who::72:2:0000-who(0000):/:/bin/who
me:3Uuv4MJmeOKhk:110:110:Sven Svensson:/usr/sven:
carl:rrFbrSmdXJoOs:111:110:Carl Carlson:/usr/carl:
```

Each line consists of a number of fields separated by a colon(:). The fields residing in the password file are displayed in the figure below.



If all information about the user/users are correct just continue with the next command. If the file contains anything that needs to be corrected this may be done using the editor **siv**. Among other things special codes may be inserted to limit the use of a password. See SA and UM.

4.5 Removing a user - rmuser

When a user no longer shall be attached to the system the command **rmuser** is used to remove him. By this command the files /etc/passwd and /etc/group are updated and the user's home directory is deleted. The home directory will not be deleted if not empty. However, deletion may be forced if the option -r is given to the command. For further information about **rmuser**, see UM.

Removing a user

```
/etc/rmuser <RETURN>
User name:
```

/etc	The command is to be found in this directory.
rmuser	The command itself.
User-name	Enter the name to be removed. The home directory is deleted if empty.

4.6 Adding new terminal/modem connections

The system is normally delivered with serial connectors for four terminals and one printer. It is possible to add more terminal-, printer- or modem connectors by installing terminal concentrator cards with eight extra connectors each.

Other types of adapters can be used for connecting terminals, e.g. Data-Board cards but they have other driver routines and are not described here.

Information about terminals from where login shall be possible and what baudrate they use is stored in the files **/etc/inittab** and **/etc/gettydefs.** In the directory **/dev** the corresponding physical units are defined. Detailed descriptions for these files are found in **SA**.

At delivery up to 12 serial connections are defined. These shall be activated in /etc/inittab at installation of the cards as in **example 1**.

Modem connections with login possibility shall also be activated in /etc/inittab and /etc/gettydefs. See **example 2**.

NOTE! After activation of the serial connections in /etc/inittab the command **'telinit q'** must be given in order for the system to read the modified file. Alternatively you can go down to single-user level with the command **'/etc/shutdown'**, whereafter /etc/inittab is read by the system when returning to multi user-level with **'telinit 2'**.

Note that printer connections, as well as eventual modem connections without login possibilities, shall not be activated in /etc/inittab. These connections however, shall be defined in the directory /dev.

Printout of the contents in the file /etc/inittab is done with the command **cat**.

Because of limited line length in the manual, long lines are divided in the listing below and on the next page. On the screen they may appear differently.

```
cat /etc/inittab <RETURN>

 is:2:initdefault:
 bl::bootwait:/etc/bcheckrc </dev/console
              >/dev/console 2>&1 ; #bootlog
bc::bootwait:/etc/brc 1>/dev/console 2>&1 #bootrun
sl::wait:(rm -f /dev/syscon;ln /dev/systty
              /dev/syscon;) 1>/dev/console 2>&1
rc:2:wait:/etc/rc 1>/dev/console 2>&1 ; #run com
err::boot:/usr/lib/errdemon -c /usr/adm/errfile
              /usr/adm/errmessages
pf::powerfail:/etc/powerfail </dev/console 1>&0
               2>&0
net:2:off:nice -16 /usr/lib/net/netman -F
ra:2:off:nice -16 /usr/lib/net/racess -F
db3:2:off:nice -16 /usr/bin/dblm /mimerdbl
               /usr/mimer/sysdb/sysdb
co:123456:respawn:/etc/getty console console
t2:2:respawn:nice -16 /etc/getty tty02 9600
t3:2:respawn:nice -16 /etc/getty tty03 9600
t4:2:off:nice -16 /etc/getty tty04 9600
t5:2:off:nice -16 /etc/getty tty05 9600
t6:2:off:nice -16 /etc/getty tty06 9600
t7:2:off:nice -16 /etc/getty tty07 9600
t8:2:off:nice -16 /etc/getty tty08 9600
t9:2:off:nice -16 /etc/getty tty09 9600
t10:2:off:nice -16 /etc/getty tty10 9600
t11:2:off:nice -16 /etc/getty tty11 9600

cat             The command cat.
/etc/inittab    The file name.
```

The remaining text is a print out of the contents in the file. 'console' is the main console and 'ttynn' indicates the remaining terminal connectors.

In this case we are interested only in the lines with **/etc/getty**. With this command the login procedure will start on the given physical units (console, tty02, tty03,...tty11). In addition to the physical unit, also a string is given as parameter, pointing at a table entry in the file /etc/gettydefs. Active terminals are marked with the text **'respawn'** which tells that the system starts login automatically on the terminal both at the first system start and when someone has logged out.

The file /etc/gettydefs can also be listed with the command **cat**:

```
cat /etc/gettydefs <RETURN>

console# B9600 HUPCL PARENB CS7 OPOST ONLCR CLOCAL
       # B9600 SANE IXANY TAB3 ECHOE CLOCAL
       #Console Login: #console

1200# B1200 HUPCL PARENB CS7 CLOCAL # B1200 SANE
      IXANY TAB3 ECHOE CLOCAL #login: #300

300# B300 HUPCL PARENB CS7 CLOCAL # B300 SANE IXANY
     TAB3 ECHOE CLOCAL #login: #1200

9600# B9600 HUPCL PARENB CS7 CLOCAL # B9600 SANE
      IXANY TAB3 ECHOE CLOCAL #login: #9600

2400# B2400 CLOCAL # B2400 SANE IXANY TAB3 ECHOE
      CLOCAL #login: #300

19200# B19200 HUPCL PARENB CS7 CLOCAL # B19200 sane
       IXANY TAB3 ECHOE CLOCAL #Login: #19200
```

```
cat              The command cat.
/etc/gettydefs   The file name.
```

The remaining text is a printout of the contents of the file.

When comparing these two files we see that only the main console tty02 and tty03 are activated in /etc/inittab, while there are rows available also for tty04..tty11. We also see that the file /etc/gettydefs tells that terminals with the parameter '9600' in /etc/inittab shall have 9600 Baud, 7 data bits and even parity (See **stty** in **UM** for a description of the parameters). The parameters in /etc/gettydefs may not contain hyphens.

Printout of the physical terminal connections in the /dev directory is done with the command **ls**.

```
ls /dev/tty?* /dev/console
/dev/console
/dev/tty02
/dev/tty03
/dev/tty04
/dev/tty05
/dev/tty06
/dev/tty07
/dev/tty08
/dev/tty09
/dev/tty10
/dev/tty11
```

```
ls              The ls command.
/dev/console    The main console.
/dev/tty?*      This parameter enables listing
                only of terminal connections.
```

The remaining text is printed out by the command. Terminal connections up to tty11 are defined.

Example 1: Activation of the terminal connection tty04.

If one extra internal terminal expansion card is installed, only the file /etc/inittab needs modifications activate the wanted connections, in this case only tty04.

If /etc/inittab does not contain the corresponding line it should be added, but in this example it already exists and shall be activated. Modify in /etc/inittab but first make a copy according to the previous instructions.

```
siv /etc/inittab <RETURN>
```

```
siv             The editor
/etc/inittab    The file to be edited
```

The entire file is now on the screen according to the previous example. The lines refering to the terminal connections console and tty02..tty07 look as follows. Console, tty02 and tty03 are already activated.

```
co:123456:respawn:/etc/getty console console
t2:2:respawn:nice -16 /etc/getty tty02 9600
t3:2:respawn:nice -16 /etc/getty tty03 9600
t4:2:off:nice -16 /etc/getty tty04 9600
t5:2:off:nice -16 /etc/getty tty05 9600
t6:2:off:nice -16 /etc/getty tty06 9600
t7:2:off:nice -16 /etc/getty tty07 9600
```

* Move the cursor to the row with tty04 and there to position 6 using arrow keys, i.e. to the text "off". If the arrow keys are not correctly set up, the control keys described in appendix A are used. (Down:CTRL-N, Up:CTRL-P, Forward: CTRL-F, Backwards: CTRL-B).

* Remove the string "off" with CTRL-D.

* Enter the text "respawn". Compare with the line for tty03.

* The speed of the new terminal (baudrate) is here assumed to be 9600 Baud according to standard. If not, the last parameter on the line may be modified to refer to another table in the file /etc/gettydefs, e.g 2400. The text string is only a pointer to the table accually, but normally the baudrate is used as pointer-string. Compare the contents of the file /etc/gettydefs, so that the correct pointer-string is used.

* Eventually, repeat this performance on the lines with tty05, tty06 etc. if more terminals should be activated.

* If anything goes wrong, **siv** is terminated with CTRL-C.

* Press CTRL-X and CTRL-F and the file will be saved.

For a simple check to see if the file has been updated use the command **cat**.

```
cat /etc/inittab <RETURN>
```

The complete file is displayed. Below is shown only the lines above, but with the terminal connection tty04 activated.

```
co:123456:respawn:/etc/getty console console
t2:2:respawn:nice -16 /etc/getty tty02 9600
t3:2:respawn:nice -16 /etc/getty tty03 9600
t4:2:respawn:nice -16 /etc/getty tty04 9600
t5:2:off:nice -16 /etc/getty tty05 9600
t6:2:off:nice -16 /etc/getty tty06 9600
t7:2:off:nice -16 /etc/getty tty07 9600
```

NOTE!! For the system to read the modified file /etc/inittab, the following command must be given. Then the activated terminal can be used as login-terminal.

```
telinit q
```

Example 2: Modem connection on the serial channel tty05.

In the first case we assume that the serial channel tty05 shall be connected to an external modem (300 Baud) with login possibility.

The serial channel tty05 shall be activated with **siv** in the file /etc/-inittab according to example 1 above, if not already active. In addition, the communication parameters are selected by defining a suitable pointer in /etc/inittab. The pointer points at a entry in the file /etc/-gettydefs. For example the pointer 300 is defined with 300 baud with the standard used in /etc/gettydefs at delivery.

"Split speed" with different transmit and receive speeds e.g. 75/1200 or 1200/75 baud) can not be used on internal tty-ports. In this case use the 5124 modem module or the 4118 module with an external modem. Alternatively the TC4204 terminal concentrator can be used. Se **SA**.

The corresponding entry in /etc/gettydefs must not contain the parameter CLOCAL if the modem signals (DCD/CTS) shall be detected. This parameter shall be removed on this entry in /etc/gettydefs for modem, requiring the DCD/CTS signal protocol.

Note! To use modem signals, the correct cable must be used (chapter 6).

After modification, the file /etc/inittab will contain the following line for the device /dev/tty05. For the rest of the file, see the previous example. Use the command **'cat /etc/inittab'** to display the file. The modified strings are marked bold below.

```
t5:2:respawn:nice -16 /etc/getty tty05 300
```

The file /etc/gettydefs is changed with '**siv**':

```
siv /etc/gettydefs
```

The line with '300' looks like this (in the manual shown on two rows):

```
300# B300 HUPCL PARENB CS7 CLOCAL # B300 SANE IXANY
     TAB3 ECHOE CLOCAL #login: #1200
```

Move the cursor to this line and to the text string 'CLOCAL'. Remove it with a few CTRL-D commands. Move to the next 'CLOCAL' and remove also this.

NOTE!!
For the system to read the modified file /etc/inittab, the following command must be given. Thereafter a user may call the modem channel from outside and login as a regular user.

```
telinit q
```

In the second case, we assume that the serial channel tty05 shall be connected to an external modem (300 Baud), but without possibility for login. The **cu** command is used to call out through the modem.

In this case the tty port must not be active in the file /etc/inittab. The corresponding line shall be marked 'off' in /etc/inittab.

```
t5:2:off:nice -16 /etc/getty tty05 9600
```

The baudrate and possibly modem signal detection are set up by parameters to the **cu** command, The option -m should be used with **cu** to activate modem signals. This enables automatic hangup if the carrier is lost. If a modem with automatic dialling is used, the -m option is normally required to achieve correct handshaking.

Also the user terminal should use modem signals (if the terminal supports these) to enable automatic hangup and exit from the **cu** command if the user forgets to close down the connection before the terminal power is shut off. Therefore give the command '**stty** -clocal' before the **cu** command.

Note! To use modem signals, the correct cable must be used; the input signals DCD and CTS are required from the external unit. See chapter 6. When dialing manually, using **cu** without the -m option, also a cable without the DCD/CTS-signals can be used.

4.7 Printout on printer

When delivered, the system is generated to support one printer port, marked PRINTER. The physical device directed to the printer port is named /dev/lp.

The program **lp** is used to queue printouts to the printer.

```
Example:    lp /etc/motd          (With the lp-system)
```

Now the contents in the file /etc/motd are printed to the standard system printer, which normally is connected to /dev/lp.

The command **lp** prints to the printer by a spooler system, but there is another spooler-system also to which printing is done with the command **lpr**.

```
Example:    lpr /etc/motd      (With the lpr-system)
```

Normally only one of these two systems is used, and the **lp**-system is as standard activated at delivery. Before using the **lp** command, the following command must be given to allow printing with the **lp** command. Log in as root or lp to give this command:

```
/usr/lib/accept  main
```

If the **lpr**-system shall be used, some changes are needed in the file /etc/rc and the file /bin/print should be modified to use **lpr** instead of **lp**. See later on in this chapter.

It is possible to activate several printers within the system. These can, for instance, be connected to terminal connectors, which may not be active in the file /etc/inittab. If new serial channels are to be used these can be created by the command **/etc/mknod** with a device name, e.g. /dev/laser or similar not starting with ttyxx.

In the command **lp** or **lpr** a "logical printer" may be defined as a parameter to which the printout shall be directed. The name of this "logical printer" is connected to a certain physical printer. If no "logical printer" is defined in the command, the standard printer for the system, normally /dev/lp, will be used.

When printing to printer with the command **lp,** the text will first be treated by a program in a command file (shell-script) before it is output to the printer. The parameter "logical printer" tells which command file to use.

When printing with the **lpr** command, the printout may be preceeded by user defined header pages and/or initiation sequences automatically taken from files, selected by the parameter "logical printer".

Descriptions of the commands used to configure the printer system can be found in **SA** and **UM**.

Changing the baudrate to a printer

As standard, communication with the printer is done with 9600 Baud. Several printers are not equipped for handling this transmission speed and therefore the speed must be changed for these printers. This is done by inserting a suitable **stty**-command in the command file for the "logical printer" in use, but may also be done directly for the physical device by the command **/etc/setspeed,** which requires super-user privileges. This command can be appended to the file /etc/rc for an automatic setup of the speed when the system is started. **setspeed** can change other parameters too, e.g. the number of data bits and parity. See **SA.**

NOTE! If the command file for the logical printer has **stty** commands, the **setspeed** command cannot be used.

Example: Changing speed to 1200 baud.

```
nice -19 /etc/setspeed -h 1200 </dev/lp &
```

Below follows a description of how this command line is appended to the file /etc/rc.

The file /etc/rc has commands that are automatically executed at the start of the system before any login is made. The outlook of the file may be different but the instructions below are independent of the exact outlook.

NOTE! You must be sure that the modification you want to make in /etc/rc is correct. Otherwise the function of the system may be violated. If anything goes wrong while modifying, the editor **siv** may be cancelled by giving CTRL-C and answer yes. See appendix A for information about **siv**.

Start **siv** from the root directory with the command below. The contents of the file will then be displayed, and may look as follows.

```
siv /etc/rc
```

```
TZ=`cat /etc/timezone`
export TZ
set `who -r`
if [ $7 = 2 ]
then
  echo "DataIndustrier DIAB AB "`uname -svr`" Virtual"
date
/etc/devnm / | /etc/setmnt
rm -rf /tmp/* /usr/tmp/*
```

```
if [ -r /usr/lib/tc/tcpar4204.tc0 -a -c /dev/tc0 ]
then
/usr/lib/tc/tcboot -s -d /dev/tc0 /usr/lib/tc/tc4204 \
                        /usr/lib/tc/tcpar4204.tc0
fi
if [ -r /usr/lib/tc/tcpar4204.tcl -a -c /dev/tcl ]
then
 /usr/lib/tc/tcboot -s -d /dev/tcl /usr/lib/tc/tc4204 \
                        /usr/lib/tc/tcpar4204.tcl
fi
if [ -r /usr/lib/tc/tcpar4204.tc2 -a -c /dev/tc2 ]
then
 /usr/lib/tc/tcboot -s -d /dev/tc2 /usr/lib/tc/tc4204 \
                        /usr/lib/tc/tcpar4204.tc2
fi
if [ -f /usr/lib/cron/log ]
then
  mv /usr/lib/cron/log /usr/lib/cron/OLDlog
  > /usr/lib/cron/log
fi
rm -f /usr/spool/uucp/LCK*
rm -f /usr/adm/acct/nite/lock*
mv /usr/adm/sulog /usr/adm/OLDsulog >/dev/null 2>&1
if [ "`uname -n`" != '(empty)' ]
then
  uname -n >/etc/systemid
  chmod 644 /etc/systemid
fi
if [ -x /etc/cron ]
then
  n=`ps lax | fgrep cron | wc -l`
  if [ $n = 0 ]
  then
    nice -16 /etc/cron
    echo "CRON started"
  fi
fi
n=`ps lax | fgrep lpsched | wc -l`
if [ $n = 0 ]
then
  rm -f /usr/spool/lp/SCHEDLOCK
  nice -20 /usr/lib/lpsched
  echo "LP scheduler started"
fi
  # /bin/su - adm -c /usr/lib/acct/startup
fi
```

The line in the example on previous page shall be inserted in the file. The line may be inserted in the file /etc/rc directly after eventual lines stating the startup of the terminal concentrator (**tc**), but before the start of the spooler system.

Move down to the appropriate line with several CTRL-N (or downward arrow). Give <RETURN> to open up a new line and move up to the new line with CTRL-P (or upward arrow). Then enter the new command. If everything looks all right according to below, the file is saved with CTRL-X CTRL-F.

The contents should then be listed with **cat** for inspection. The file should be as listed below. However, only the beginning of the file and its inserted line with **setspeed** is shown below:

```
    cat /etc/rc

TZ=`cat /etc/timezone`
export TZ
set `who -r`
if [ $7 = 2 ]
then
  echo "DataIndustrier DIAB AB "`uname -svr`" Virtual"
  date
  /etc/devnm / | /etc/setmnt
  rm -rf /tmp/* /usr/tmp/*
  if [ -r /usr/lib/tc/tcpar4204.tc0 -a -c /dev/tc0 ]
then
/usr/lib/tc/tcboot -s -d /dev/tc0 /usr/lib/tc/tc4204 \
                         /usr/lib/tc/tcpar4204.tc0
fi
if [ -r /usr/lib/tc/tcpar4204.tc0 -a -c /dev/tc1 ]
then
/usr/lib/tc/tcboot -s -d /dev/tc1 /usr/lib/tc/tc4204 \
                         /usr/lib/tc/tcpar4204.tc1
fi
if [ -r /usr/lib/tc/tcpar4204.tc2 -a -c /dev/tc2 ]
then
/usr/lib/tc/tcboot -s -d /dev/tc2 /usr/lib/tc/tc4204 \
                         /usr/lib/tc/tcpar4204.tc2
fi
nice -19 /etc/setspeed -h 1200 </dev/lp  &

    ETC. the rest of the file /etc/rc !
```

To have the function activated through the file /etc/rc the system must be taken down to the single-user mode with '**/etc/shutdown**', and restarted with the command '**telinit 2**'.

Activating **lpr** instead of **lp.**

If the print-spooler system **lpr** shall be used instead of the **lp**-system, the file /etc/rc and the command file /bin/print must be modified. This is done with the **siv** command.

```
siv /etc/rc
```

Now /etc/rc is displayed on the screen according to the previous pages! The lines to be modified are at the end of the file and look as follows when the **lp**-system is active:

```
n=`ps lax | fgrep lpsched | wc -l`
if [ $n = 0 ]
then
  rm -f /usr/spool/lp/SCHEDLOCK
  nice -20 /usr/lib/lpsched
  echo "LP scheduler started"
fi
```

These lines should be marked with the character # at the beginning of each line to indicate that they are comment lines only. After them the following lines shall be appended to the file.

```
rm -f /usr/spool/lpd/lock /usr/spool/lpd/lpdctl
rm -f /usr/spool/lock/*  /usr/spool/lpd/ERRLOG
nice -20 /usr/lib/lpd &
echo "LPR spooler started"
```

The end of the file /etc/rc will then contain the following text where the changes are marked in bold face.

```
#  n=`ps lax | fgrep lpsched | wc -l`
#  if [ $n = 0 ]
#  then
#    rm -f /usr/spool/lp/SCHEDLOCK
#    nice -20 /usr/lib/lpsched
#    echo "LP scheduler started"
#  fi
   rm -f /usr/spool/lpd/lock /usr/spool/lpd/lpdctl
   rm -f /usr/spool/lock/*  /usr/spool/lpd/ERRLOG
   nice -20 /usr/lib/lpd &
   echo "LPR spooler started"
  # /bin/su - adm -c /usr/lib/acct/startup
fi
```

After this, the command file /bin/print should be modified to call **lpr** instead of **lp**. This is also done with **siv**.

```
pr -151 -b -w112 $* | lp    (With the lp-system)

pr -151 -b -w112 $* | lpr   (With the lpr-system)
```

To have the function activated through the file /etc/rc the system must be taken down to single-user mode with '**/etc/shutdown**', and then restarted with the command '**telenit 2**'.

4.8 Parameters for terminals, modems and printers

This section gives a summary of different factors to be observed at connection of terminals, modems or printers to normal serial connectors in the system, on the computer module or by the internal terminal expansion cards. These connectors are all handled by the same driver program in the operativ system.

For additional terminals and printers connected by DataBoard cards or TC cards, other driver routines in the system are used. For these, separate descriptions are available. See also **SA**.

Programs and system files mentioned here are described in detail in other sections of this manual or in **SA**.

Connection of devices to serial ports

When connecting different types of terminals, modems and printers to the system, a serial interface V.24/V.28(RS232C) with asynchronous communication is used.

One of the ports is always the main system console (/dev/console) and another one always used for system printer (/dev/lp). Other ports may be defined as required by the user. As console in single user mode the device /dev/syscon is used, which normally is linked to /dev/console.

When an external device is connected it is in general adapted to the computer by setting communication speed (baudrate) and other parameters locally through strapping or local commands. The table below shows the parameters normally used by the system.

	Terminal	Printer
Baudrate	9600 Baud	9600 Baud
Interface	V.24(RS232C)	V.24(RS232C)
Wordlength	7 databits	7 databits Note 1)
Parity	Even parity	Even parity
Stopbits	1 stopbit	1 stopbit
Buffer-full-signal	XON/XOFF	XON/XOFF Note 2)
Terminal type	VT100	Note 3)

Note 1) Terminals and printers may also be set to 8 databits, no parity, if the parity bit (the 8:th bit) from the computer is ignored by the equipment. As standard, the computer will just ignore eventual incomming parity bits.

Note 2) The system printer (/dev/lp) also maintain static buffer-full signalling from the printer, by requiring an active high level on the computer's CTS port to allow data to be sent to the printer. When using the standard cable 6464-00, this signal is from pin 19 on the 25-pin printer connector. For other printers, this signal might be sent from another pin, requiring a change in the cable.

Note 3) The terminal type (TERM) is normally automatically defined at login by the commands in the user's file .profile.

Login terminal and modem with login ability

In multi-user mode (normal at automatic start-up) baudrate and other communication parameters are set for the terminals during the login procedure, as stated in the files /etc/inittab and /etc/gettydefs. While the user tries to enter his user name during login, the system tries to adjust the terminal parameters to the input characters. See the previous example and **SA**. After login some parameters are normally changed by a **stty**-command in the file .profile.

The terminal parameters may be changed temporarily after login with the command **stty**, but if the communication speed or similar parameters are changed this requires the terminal to be locally readjusted after login and after the command **stty** is given. See **stty** in **UM**.

After login, the commands in the file **/etc/profile** and in the file **.profile** in the user's home directory, are automatically executed. The file .profile is automatically created by **/etc/mkuser** when a new user is defined and a new directory is assigned as login directory. In .profile, the terminal type and certain control characters are defined for the terminal.

At manual system start, the file /etc/inittab is not used until the multi-user mode (Boot level 3) is entered. The main console baudrate and terminal type for the main console, however, can be set as parameters in a circuit, **/dev/nvram**. These are defined with the command **/sas/bootpar**. See **SA**.

At manual startup, when the system has been started in single-user mode (boot level 2), the commands in /etc/profile and /.profile are performed.

The terminal type is assigned by the shell parameter **TERM,** which is defined in .profile. TERM is defined as a string and is a parameter, used in certain programs (e.g. the editor **siv**) together with information in the directory **/usr/lib/terminfo** or in the file **/etc/termcap**. There is defined which control characters are required by the terminal and other terminal depending parameters, e.g. cursor control characters and the number of rows on the screen. These parameters are used by different programs with screen handling, the **siv** editor as example. The file /etc/termcap is an older type of parameter file in text format and can be displayed on the screen with the command 'cat /etc/termcap'. The directory terminfo contains a file for each terminal in compiled form. However, at delivery both files are compatible. Note that the **siv** editor always uses the file /etc/termcap. There are also programs that use code-files of their own instead of these standard files.

The actual value of TERM can be displayed by the command:

```
echo $TERM
```

To see which terminal types are available in the system, i.e. which names the parameter TERM can be set to, the following command may be given for display of the header lines in the parameter file /etc/termcap:

```
fgrep '|' /etc/termcap
```

A selection of lines from the file will be displayed where the names of the terminal types are given. The lines contain several text strings, enclosed by the character '|'. The second text-string 'twist' in the example below indicates the value in TERM to select this terminal type. Example of a line listed with the above command (terminal type twist):

```
dl|twist|vt-100|pt100|pt-100|dec vt100:|
```

The most common terminal types are, adm3a, vt100, twist (24-row Facit Twist), twi72 (72 row Facit Twist), vt220 and vt240.

Printers and modems without login ability

For a printer all communication parameters can be changed. This can be done automatically at system start by writing the command **/etc/setspeed** in the file **/etc/rc**.

'Split speed' baudrate (1200/75 or 75/1200) can not be used for the internal terminal connections.

More printers may be connected to the system via the serial ports and texts are sent to the printer via a program system called printer spooler. The system is delivered with two different spooler systems, the **lp**-system and the **lpr**-system. Only one of these shall be active in the system.

When a modem is to be connected to the system via a serial port, this port normally shall be modified according to the example 2 above, even if login is disabled.

4.9 Initiating a new file system on the internal disc unit

The description below explains how to initiate a new internal winchester disc as root file system, from the distributed system diskettes; the BOOT diskette and the tar-diskettes.

NOTE! At delivery of a new system, the internal winchester disc is always initiated and installed with system programs.

NOTE! This description assumes that the winchester disc is formatted from the deliverer. Do never format a winchester disc without previous contact with your deliverer. Normally the formatting is not destroyed when disc errors appear. Formatting is only required after reparing failures in the read/write heads or other hardware by a service technician.

To create a new file system on the internal hard disc unit, the system shall first be started in manual mode up to single-user mode (boot level 2) with the diskette as root device according to section 3.1.2. The diskette to be used is marked:

```
Exempel:    D-NIX 5.2   072-8701-00/ 1(1)
            Version X.xx
            DS90-00 / BOOT
            mount format
            Copyright Diab Data AB
```

The main console screen should after startup look like this:

```
M68020 prom
sf(0,0)
System boot
Root device "xx(nn,nn)" ? sf(0,0)
> dnix
OS:INFO: D-NIX X.X Virtual Ver X.X -
         System xxxk  User xxxxk  Swap x.xM
#
```

The steps in the following are:

1. Create a file system on the internal wichester with **/etc/mkfs**.

Example:

```
/etc/mkfs  -b 1024 /dev/si32
```

The **mkfs** command senses the volyme size but the blocksize 1 kbytes (1024 bytes) shall be used in DS90-00 instead of the standard 2 kb.

The file system on the winchester hard disc is now initiated, but contains no files. If an external hard disc unit has been initiated like this (with another device name as argument to **mkfs**) it may now be connected to the root filesystem as a sub directory with the **mount** command.

The internal winchester disc is normally used as root filesystem, which requires system files to be copied and certain special files to be created. This is done with the program **harddoit**, which transfers data from the BOOT diskette to the hard disc and initiates the required device tables to use the hard disc as root filesystem.

NOTE! Now the Release Notice, which is delivered with the software, must be studied. There may be described deviations from the procedures below.

2. Copy the most important files from the BOOT diskettes with the **harddoit** command.

```
/harddoit
dd: 1+0 records in
dd: 1+0 records out
tar: block factor 1
```

The lines under the command are printouts from the **dd** and **tar** commands in harddoit.

The harddoit command itself and certain programs in the /sas directory on the BOOT diskette are not copied, as these are only used with the BOOT diskette as root device.

3. Close down the system and restart it manually up to single-user mode, this time from the internal winchester disc. The system must be stopped with the **kill 1** command, as the normal command **/etc/shutdown** is not available on the BOOT diskette.

```
kill 1
```

When the text **System halted** is shown on the terminal, the system is started from the newly initiated internal hard disc. Note that the system can not be started up to multi-user mode with the files now available. The key shall still be in the position MANUAL and the startup is done in manual mode up to the single-user level, now with the internal disc as root filesystem, by only giving RETURN as answer to all questions and by giving the command **dnix** from the stand-alone system, until the prompt # is achieved. The screen should then look like this:

```
M680XX prom                    (XX = 00 or 20)
                              <Give only RETURN>
> dnix
OS:INFO: D-NIX X.X Virtual Ver X.X -
         System xxxk  User xxxxk  Swap x.xM
#
```

4. Load the other system files from the tar-diskettes according to the Release Notice, delivered with the software (the **'tar -xvf /dev/sf0'** command).

The number of tar-diskettes can vary, depending on the diskette size. The diskettes shall be loaded in number order according to the subnumbers on the diskettes. First the diskettes marked UNIQUE shall be loaded, then those marked COMMON.

After loading all tar-diskettes, the command **/etc/rinstall** is given (see the Release Notice). The most important system files are delivered with file names ending with the character '+' to enable for the user to select using the new files or the old system files and to enable the user to automatically update the new operating system file with the old system parameters. (Compare the **/etc/mkcfig**) command in **UM** or **SA**). This is done with **/etc/rinstall** allowing the user to answer questions. When the internal disc has been initiated, the answer to all questions shall be 'y' (for yes).

Example of diskette labels, where XX corresponds to the computer type.

```
D-NIX 5.2   072-8701-XX/ n(2)
Version X.xx
DS90-XX / UNIQUE
tar format
Copyright Diab Data AB
```

```
D-NIX 5.2   072-8753-00/ n(4)
Version X.xx
DS90-XX / COMMON
tar format
Copyright Diab Data AB
```

where /n is /1, /2, etc.

5. After loading all diskettes and executing the **/etc/rinstall** command according to the Release Notice, the system shall be closed down and restarted again, now in AUTO mode. The system shall be configured for the application environment by defining new users and passwords and other parameters. The printer spooler shall be aktivated. To do all this the key is first turned carefully to the AUTO position, Thereafter the following command is given:

```
/etc/shutdown -k
```

When the text **System halted** is seen on the terminal, the system will automatically start up to multi-user mode. If the system has been stopped with the key still in the MANUAL position, the message 'M68000 prom' will appear on the screen. In this case press the button INIT, with the key in MANUAL, and then turn the key to AUTO to initiate the automatic start sequence.

6. Now a total backup of the entire main system should be taken by copying the entire disc to a tape streamer (or to diskettes).

7. Thereafter all optional programs are loaded and possibly earlier saved user files from a backup, whereafter a new total backup should be taken, with the system ready for use.

Example of a **harddoit** file:

Below is one example of harddoit, for the DS90-00.

The long line with 'tar' is shown below divided, due to the limited line length in the manual. The character '\' (upper case \) at the end of the line indicates that the line is divided. On the screen it may look different.

```
# harddoit boot routine

MNT=/mnt
DEV=/dev/si32

dd if=/dev/sf0 of=$DEV count=1

if [ `/etc/mount $DEV $MNT` ]
then
  echo "harddoit: ERROR: Failed to mount $DEV"
else
  # Copy everything
  tar cfS - bin usr etc dev tmp sas sf0 .profile \
                dnix boot | (cd $MNT; tar xf -)

  # Make an empty /etc/mnttab
  >$MNT/etc/mnttab

  /etc/umount $DEV
fi
```

5. Maintenance

5.1 Preventative maintenance

5.1.1 Files and directories in the file system
5.1.2 Copy
5.1.3 Testing filesystems on disc units - fsck

5.2 Security copying, Backup

5.2.1 Copying own files and directories on diskette - copy
5.2.2 Copying of all user directories
5.2.3 Copying of a complete system (total backup)

5.3 System messages

5.1 Preventative maintenance

5.1.1 Files and directories in the file system

After login, the super-user will be in the root directory. A directory is a file that contains a list of file names and subdirectories together with information about these files and subdirectories. Each subdirectory in turn, also contains a list of file names and subdirectories plus information about their location within the file system. Generally, the system can have an arbitrary number of levels with subdirectories. When the command **l** is given, the contents of current directory will be printed on the screen.

If nothing else is stated below, it is assumed that the user is in the root-directory. The contents of the root directory is printed out if the following command is given:

* Enter **l** (Note: lower case L), press <RETURN>.

This is an example of a listing of a root directory (the actual contents may vary)

```
total 740
drwxrxr-x   2 bin    bin     168 Oct 27 09:13 bin
-rwx-----   1 root   sys     749 Sep 29 09:54 boot
drwxr-xr-x  3 root   sys     880 Oct 27 08:14 dev
-rw-r--r--  1 root   sys     151 Oct 20 16:01 dnix
drwxr-xr-x  2 root   sys     688 Oct 27 08:57 etc
drwxrwxrwx  3 root   root     80 Oct 23 09:36 hnd
drwxrwxr-x  2 bin    bin      32 Oct 22 13:27 lib
drwxrwxrwx  2 root   root    048 Oct 16 17:56 lost+found
drwxrwxrwx  2 root   sys      32 Oct 27 10:58 sf0
drwxrwxrwx  2 root   sys      32 Oct 27 12:08 mnt
drwxrwxr-x  3 root   root     80 Oct 22 13:26 sas
drwxrwxrwx  2 root   sys     400 Oct 27 09:41 tmp
drwxr-xr-x 10 root   sys     160 Oct 22 13:43 usr
```

drwxr-xr-x	User permission modes for a file or directory. Described on the following pages.
root	Owner of the directory usr
sys	Group of the directory usr
160	Number of bytes in directory usr
Oct 22	Date of last modification of usr
13:43	Time of last modification of usr
usr	Name of the directory usr
. . .	Corresponding information about other files/directories

File types

Each line holds information about a file or a directory.

The first character on the line states whether it is a file or a directory, for example:

```
drwxrwxr-x  2 bin   bin   1168 Oct 27 09:13 bin
```

The character d states that this is a directory. A file is denoted like this:

```
-rw-------  1 root   sys  51749 Sep 29 09:54 boot
```

The character "-" in the first position states that this is an ordinary file.

Owner/Group/Others

Each file or directory has an owner. In most cases, the person who created it has the ownership. The owner of a file or directory may give it different types of protection.

The system has three different types of owners to a file or directory:

owner (user)	The person who created the file or directory.
group	A number of users may form a user group. Each file or directory can be owned by a group.
others	Others includes all other users in the system. Note that this is not limited to the default group with the name 'other'.

I.e. the system has three kinds of owners.

user - group - others

Each user belongs to a certain group which is assigned when the user is defined (See chapter 4).

File protection, access permissions

In the file system, each file and directory can have three different types of conditions describing what can be done with the file/directory. As a file differs slightly from a directory, the protection types have different meanings. These protection types are sometimes called either file permissions, access permissions, access modes or access codes.

Read
r

A user with read access to a file may read its contents.

A user with read access to a directory can list the files in the directory.

Write
w

A user with write access may change the contents in this file. Note that this is valid even if the user has no write access in the directory where the file is located.

A user with write access in a directory may change the contents of the directory, i.e. create new files, delete old files.

Execute
x

A user with execute access may use the file name as a system command.

A user with execute access to a directory may change from current directory to this directory. The user may also copy files from this directory provided that the user has read access to the directory.

Each of the owner types may have one, two or all three types of access permissions. This means that there is a total of 9 different access permissions for a file.

Owner	Group	Others
rwx	rwx	rwx

If access is not permitted, the character "-" is shown at the corresponding positions, meaning the file/directory is protected. The nine access permissions are called the access modes of a file or directory. For example:

```
drwxr-xr-x  6 Karl dok   2144 Oct 23 06:17 /usr/Karl
```

The owner (in this case 'Karl', has read, write and execute permissions for the directory, while the group 'dok' and other users in the system only have read and execute permissions.

For the file boot the following is valid:

```
-rwx------  1 root sys  51749 Oct 16 14:03 boot
```

The owner of this file has read and write permissions and also permission to execute the file as a command. The group 'sys' and others have no access permissions at all and can therefore neither read, nor write or execute the file.

Special files

The directory **/dev** does not contain normal files. The "files" in /dev are called special files and are used only for direct access to the driver routines in the operating system for different physical units. These are described in SA.

New special files can be added with the command **/etc/mknod** and only if the corresponding driver routine is available in the operating sys tem. Special files can be deleted with the command **rm** and linked to each other with the command **ln**.

5.1.2 Copy

The contents of one or several directories may be copied to another directory or to a file system on another media. In order to do this, the command **copy** is used. With **copy**, the tree structure of the file system, from where the **copy** is made, is preserved.

Copy all files in a directory to a mounted 8 inch diskette.

```
copy -mrvta /usr/mydir /sf0/usr/
```

copy	The command itself.
-m	The files that are copied will have the same access time and modification time as the original files.
-r	All directories that are found will be searched through.
-v	There will be a list of filenames printed on the screen when the copying is performed.
-t	Preserves the tree structure in the directories that are being copied, here /dev/mydir.
-a	Before each file is copied, the user must accept it by pressing 'y' for (yes).
/usr/mydir	The name of the directory that shall be copied.
/sf0/usr	States to which media the directory is to be copied and under which directory it should be saved. In /sf0/usr, the directory mydir is created if not already existing and all files are copied.

5.1.3 Testing file systems on disc units - fsck

This command should be used by the system administrator exclusively and only on disk units not mounted in the system. The command shall only be used in the stand-alone system (boot level 1 after a manual start) when the internal winchester disc shall be tested. See **SA**.

File systems on both diskettes and winchester disks, can be tested with the command **fsck**. Normally no data is destroyed.

If the system discovers errors in a file system, it will be reported. With the command **fsck** it is possible to try to repair a damaged file structure.

Test of the root filesystem on the internal winchester disc

First close down the system and make a manual start up to the stand-alone system (boot level 1) according to chapter 3.1.2. (Do not start 'dnix'). The start can be with the internal disc as root file system.

```
>/etc/fsck -rr si(32,0)

Checking si(32,0)
File system: root                Volume:
Volume size =   82Mb,
Block size = 2048, 13312 l-nodes (13312 cont.)

    Phase 1 - Check l-nodes
    Phase 2 - Check Pathnames
    Phase 3 - Check Unreferenced files
    Phase 4 - Check Link Counts
    Phase 5 - Check Bitmap

    Total number of files:    662
    Total number of blocks: 41472,  37136 - free blocks
```

/etc/fsck	The command itself.
-rr	Shall always be stated in stand-alone-mode.
si(32,0)	The device on which the file system is. Shall be defined in the same form as at manual start up from the device.
Remaining text	Is printed out by the system to inform what is being checked.

If the command **fsck** detects files or directories that are not connected to the file system, these are linked to the /lost+found directory at restoration. In this directory you may find data that has been lost.

If the system is started manually and the message "Warning - File system not clean" appears, 'dnix' must not be started. In this case the root filesystem shall first be tested with the **fsck** command. If the system was started from the BOOT diskette, the warning message will appear if the diskette earlier was removed from the system without being correctly closed. **fsck** can easily correct this. The unit name for the BOOT diskette is normally sf(0,0). Normally test the BOOT diskette with:

```
>/etc/fsck -rr sf(0,0)
```

Testing a file system which is not mounted

A file system, which is not mounted in the system, can be tested from the operating system (single- och multi-user level), but in this case the device name shall be given as argument to **fsck** instead of the unit name above. For testing a file system on a diskette, give the following command:

```
/etc/fsck  /dev/sf0
```

The **fsck** utility is described in more detail in **SA** and **UM**.

5.2 Security copying, Backup

It is important to make sure that there are copies, backups, of all files, in order to restore the system if, for example, a file unintentionally should be deleted.

Backup should be done regularly by the system administrator or other person responsible for the system. In many cases the user himself wants to make copies of his own files, maybe as backup, or for saving earlier versions of files or for transport of files to other computers, for example by using diskette or tape.

The command **tar** saves files to and restores them from an external media, often called 'archives', in a special standardized format which is independent of the physical file structure in the file system. Suitable media are streamer cassettes and diskettes. When using **tar**, the diskettes shall only be formatted in a standard format. See the command **format**.

The files are copied directly to the physical media with a block containing information about the original location of the files within the directory structure, file access permissions, etc. in the file system from which the copying is done. The information is sufficient for the **tar** command to reconstruct the copied files or directories. Selected files or all files and directories can be read back to the system when needed.

NOTE! It is recommended that files normally are copied without entering / (slash) preceeding the file name. Instead the user should move (**cd**) to the directory where the files reside before the **tar** command is given. In this way it is possible to read back the files and directories to a temporary directory without deleting the original files on the disk. Alternatively, the option -**A** may be given to the **tar** command when restoring the files. This option removes any / in the beginning of the file names.

When reading back saved files to another computer system, the user and group name should be changed with the option -**o** added to the **tar** command as **tar** only saves this information in numeric form.

See **SA** and **DM** for further information about the 'backup' utilities.

If the backup is done with the **tar** command in the background the exit status from **tar** should be tested to verify that the copying was OK. This can be done with the **test** command. The example below shows copying to streamer The **tar** command is explained later in this chapter. These commands are suitable to place in a file to be executed as a shell procedure in the background (with &) causing the message "Error at backup" to be given on the main console if the backup failed. Eventual error messages are placed in the temporary file tmp/tartmp for later inspection. The used commands are described in **UM**.

Example:

Backup in the background of the complete /usr directory to a streamer tape with testing. The commands below is in a command file which is started in the background.

```
cd /
tar -cfbk /dev/st0 200 60300 usr >tmp/tartmp 2>&1
if test $? -ne 0 ; then
   echo "Error at backup" >/dev/console
   exit 1
fi
```

The **cron** command can be used for automatic backup, e.g during night. Then it would be suitable to testread the copy to get a thorough check. The following commands should then be executed after the backup.

Example:

Test commands to be inserted after the commands above when automatic backup is used.

```
tar -tf /dev/st0 >/dev/null 2>&1
if test $? -ne 0 ; then
   echo "Error at readtest of backup" >/dev/console
   exit 1
fi
```

5.2.1 Copying own files and directories to diskette

Backup to diskette

One or several own files and directories can be saved on diskette with the following routines. A formatted 8 inch diskette is placed in the diskette unit. Eventual data on the diskette will be deleted.

```
tar -cvfk /dev/sf0 980 myfile <RETURN>
```

tar	The command itself.
c	States that a new tar-diskette shall be created. c = write to given unit.
v	Prints a list of what happens during the process to standard output, in this case on the screen.
f	States that the next argument is the name of the device to which copying will be done.
k	States that next argument (980 in the example) is the maximum storage capacity.
/dev/sf0	Name of the device to which the backup will be done.
980	This shall be the real size of the diskette, expressed in Kbytes.
myfile	Directory or file to be copied. One or seve-givenay be entered.

If the diskette becomes full, a message is displayed on the screen and the operator may change to a new diskette to continue the copying. At a later reading the diskettes should be read in the same order.

The contents of a tar-diskette can be **listed** on the screen with the following command, where each file is listed on a separate line.

```
tar -tvf /dev/sf0 <RETURN>
```

Reading of a backup copy from diskette

A selected directory or file can be read back with the command **tar**. The command **cd** is given first to ensure that the files are restored to the correct directory.

```
tar xvf /dev/sf0 myfile <RETURN>
```

tar	The command itself.
x	States that reading should be done in tar-format. x = eXtract from given unit and file name.
v	Prints a list of what happens during the process to standard output, in this case on the screen.
f	States that the next argument is the name of the device from which copying will be done.
/dev/sf0	Name of the device to which backup was done earlier.
myfile	Directory or file to be copied. One or several can be entered. If no filename is given all files are copied from the diskette.

Further information on parameters to **tar**, see **UM**.

5.2.2 Backup copying of all user directories

All user directories should be regularly saved by the system administrator. This is done most conveniently to a cassette streamer if installed on the system, otherwise to diskettes.

NOTE! Super-user privileges are required for having access to all files. No other user may be logged in during the process.

Copying of the directory usr to diskette

To save on diskette the following **tar** command is given. The diskette shall be formatted. More than one diskette may be needed, in which case the operator gets a message to change diskette.

```
cd /
tar -cvfk /dev/sf0 980 usr <RETURN>
```

cd /	Move to the root directory where usr is found.
tar	The command.
c	States that a new tar-diskette shall be created. c = write to given unit.
v	Prints a list of what happens during the process to standard output, in this case on screen.
f	States that the next argument is the name of the device to which copying will be done.
k	States that the next argument (980 in the example) is the maximum storage capacity.
/dev/sf0	The name of the device to which the backup shall be done.
980	This shall be the actual size of the diskette expressed in Kbytes.
usr	The directory usr contains all user direc-willes to be copied.

Reading of all user directories from the diskette(s) is done by the command:

```
cd /
tar -xvf /dev/sf0 <RETURN>
```

If several diskettes were used, these should be read in the same order as written earlier.

Copying the directory usr to cassette streamer

To save the directory usr on a cassette streamer the following command shall be given. The example is for a 60 Mbyte streamer cassette.

```
cd /
tar -cvfbk /dev/st0 300 60300 usr <RETURN>
```

cd /	Move to root directory where usr resides.
tar	The command itself.
c	States that writing shall be done from the beginning of the cassette. c = write to given unit.
v	Prints a list of what happens during the process to standard output, in this case on the screen.
f	States that the next argument is the name of the device to which copying shall be done.
b	States that the next argument (200 in the example) is the block size.(See the comment below).
k	States that next argument (60300 in the example) is the maximum storage capacity.
/dev/st0	The name of the device to which the backup shall be done.
200	This shall be the block size at copying. The value 200 is suitable for a streamer. Default block factor is '20' if nothing else is stated which makes backup slower.
60300	This shall be the actual size of the cassette expressed in Kbytes. Here 59 Mbytes = 59*1024 Kbytes which gives some margin to 60 Mbytes.
usr	The directory usr contains all user directories to be copied.

Note that the **tar** command gives a warning-text as the block size exceeds 20. This is ok as the real block size on the tape always becomes 20 when writing to a streamer. The value 200 is used instead as buffer size by **tar** for achieving rapid data transmission.

Reading of all user directories from the cassette is done with the command below on condition that the whole cassette will be read.

```
cd /
tar -xvf /dev/st0 <RETURN>
```

5.2.3 Backup copying of a complete system (total backup)

For backup of the entire system with all its files and directories the following **tar** command is used. This should be done regularly by the system administrator. For a total backup, use of a cassette streamer is recommended. At least six diskettes are required for total backup of the basic system plus diskettes for options and for the user's files.

The read back of a complete system that has been backed up is normally done to an empty recently initialized disk memory. After installation of the disk memory and after login to the root directory as 'root' the backup will be read.

Also special files in the directory /dev may be backed up with **tar** (the option S, i.e capital S), but be aware that it is impossible to restore already existing special files when reading back from the backup copy.

NOTE! Super-user privileges are required and no other users may be logged in on the system! The best way to do is to enter the single-user level with the command **"/ect/shutdown"**.

Backup of the entire system onto diskettes

```
cd /
tar -cvfkS /dev/sf0 980 . <RETURN>
```

tar	The command itself.
c	States that a new tar diskette is to be created. c = write to given unit.
v	Prints a list of what happens during the process to standard output, in this case on screen.
f	States that the next argument is the name of the device to which copying shall be done.
k	States that the next argument (980 in the example) is the maximum storage capacity.
S	States that also special files in /dev shall be copied.
/dev/sf0	The name of the device to which the backup will be done.
980	This shall be the actual size of the diskette expressed in Kbytes.
.	All files in the entire system are copied. The argument dot (.) includes also filenames beginning with a dot.

Reading of the entire system from the diskettes is done with the command below. The diskettes must be inserted in the same order as written. A message is given when the diskettes shall be changed.

```
tar -xvf /dev/sf0 <RETURN>
```

Backup of the entire system to a cassette streamer

To save on a cassette streamer the following command shall be given. The cassette shall be rewound. The example is for a 60 Mbyte streamer cassette.

```
cd /
tar -cvfbkS /dev/st0 200 60300 . <RETURN>
```

tar	The command itself.
c	States that writing shall be done from the beginning of the cassette. c = write to given unit.
v	Prints a list of what happens during the process to standard output, in this case on the screen.
f	States that the next argument is the name of the device to which copying shall be done.
b	States that next argument (200 in the example) is the block size. (See the comment below).
k	States that the next argument (60300 in the example) is maximum storage capacity.
S	States that also special files in /dev shall be copied.
/dev/st0	The name of the device to which the backup shall be done.
200	This shall be the block size at backup. The value 200 is suitable for a streamer. If nothing else is stated '20' will be default which makes backup slower.
60300	This shall be the actual size of the cassette expressed in Kbytes. Here 59 Mbytes = 59*1024 Kbytes which gives some margin to 60 Mbytes.
.	All files in the root directory are being copied. The argument dot (.) includes also filenames beginning with dot.

Note that the **tar** command gives a warning text as the block size exceeds 20. This is ok as the real block size on the tape always becomes 20 when writing to a streamer. The value 200 is used instead as buffer size by **tar** for acheiving rapid data transmission.

Reading of the entire system from the cassette is done with the command below.

```
tar -xvf /dev/st0 <RETURN>
```

For more information on parameters to **tar**, see **UM**.

5.3 System messages

System messages with information about incidents and errors are given by the operating system directly and only to the main system console. If the log function is activated, the system messages are also gathered and saved into a file. See **errdemon** in **UM**.

These messages are given in english and in very short sentences.

Some messages are related to the hardware as for example disk units or the internal memory. In this cases, addresses and device numbers (major/minor) are often given in the message. When error messages are displayed this information should be noted.

Certain messages begin with the word **PANIC**. This means that a serious malfunction has occured and the system halts. If the key switch is in position AUTO the system will be restarted, in which case the automatic disk test normally has to correct the file system.

Below is a list with some typical error messages with very abbreviated explanations.

```
OS:  PANIC:  Trap in system mode - SYSTEM HALTED!
     Trap=nn, ssp=nn, sr=nn, ssr=nn, pc=nn, fad=nn
     d0-d7 nn nn nn nn  nn nn nn nn
     a0-a7 nn nn nn nn  nn nn nn nn
```

This message is given if the system comes to a dead stop due to an error which the operative system cannot handle without risks of destroying information. Normally the system can be restarted.

```
End of nodes
Too many active processes
End of swap area
```

These messages may be due to an incorrect user program, which requires too much sysem resources. When running applications with high demands on these parameters, the command **mkcfig** can be used to change the system. See **SA** and **UM**.

The error messages below are due to externally connected units such as, disk units, tape streamers, diskette units etc. The messages often include the device number (major/minor) for identification. See **SA**.

Write protect

Write protected. The write protection on a diskette is activated.

Device busy

Can be received at 'umount' of a diskette or a disk unit if a file is still open.

Drive off line

As example, this can indicate that a diskette or a tape is not inserted in the diskette or streamer unit.

No space on dev mm,nn
bn=nn cmd=nn sts=nn

No space on the external unit.

Timeout

The external unit does not answer within a defined minumum time.

Error on dev mm/nn
bn=nn cmd=nn sts=nn

If this type of error message is given, please not the given values of the block number(bn), the command(cmd) and the status(sts). This message could be caused by a soft error in some block on the disk unit. Normally soft errors are repaired at a new start-up as the command **fsck** is automatically executed.

6. Cables and connectors to the system

6.1 Connection of terminals and printers

6.1 Connection of terminals and printers

The basic system is delivered with five terminal connections and one printer connection. Serial asynchronous communication according to V.24/V.28 (RS232C) is assumed.

The transmission speed (baudrate) and all other parameters can be changed with software and is described in general in chapter 4 and in more detail in SA. At a manual system start, the following standard setting is always required for the terminal used as the main console terminal. This is the standard also used for printers connected to /dev/lp, if the parameters are not changed.

In single-user mode the console terminal is connected to the unit /dev/syscon instead of to /dev/console but normally linked to the same serial port. In single user mode the terminal parameters are always setup according to the file /etc/ioctl.syscon. This file is automatically created by the system when entering the single-user level mode from the multi-user mode.

Standard parameters for terminals and printers are:

Parameter	Terminal VT100	Printer
Interface	V24 (RS232C)	V24 (RS232C)
Baudrate	9600 Baud	9600 Baud
Number of databits	7 databits	7 databits
Number of stopbits	1 stopbit	1 stopbit
Parity	Even parity	Even parity

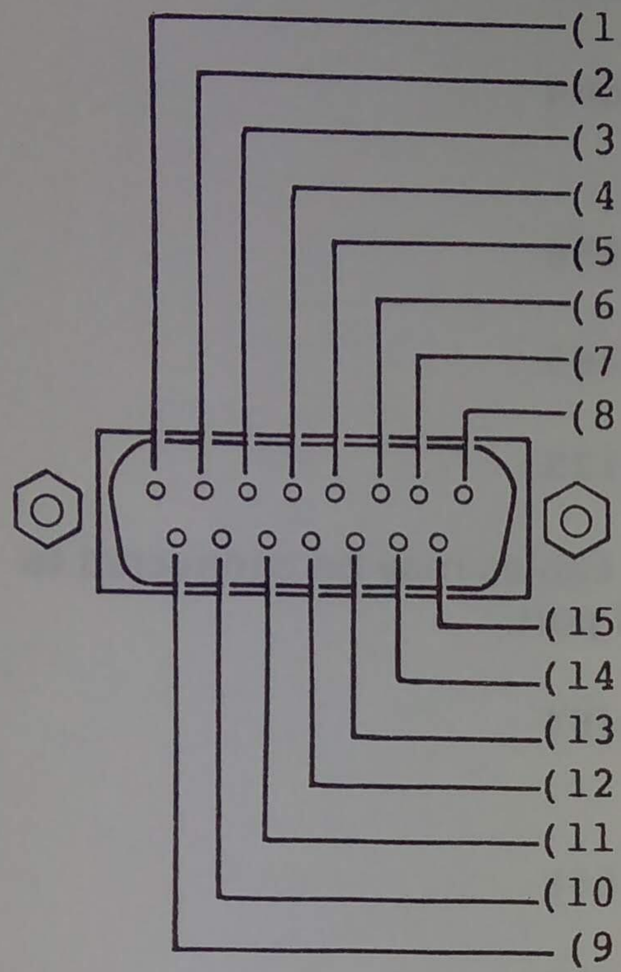
With the standard parameters, the computer ignores any incoming parity bit but sends parity in the 8:th bit. Therefore, terminals or printers ignoring any incomming 8:th data bit may also be set to:

Number of data bits	8 data bits	8 data bits
Parity	No parity	No parity

On connected terminals, however, it might be neccessary to do some local changes for complete adaption to the system and its parameters.

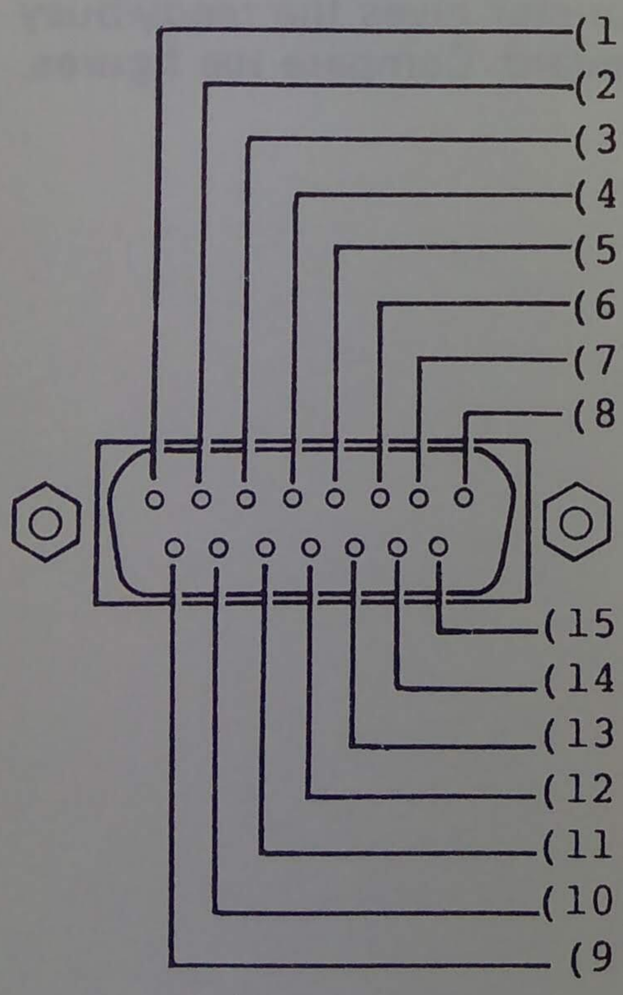
The pin layout in the serial connectors are as below. The connector type is DA15P and is seen from the outside below. There are certain differencies between the ports CONSOLE and PRINTER and the other terminal ports on optional terminal expansion modules. If any 4204 terminal concentrator is used, see **SA** or the Release Notice for TC4204.

Serial connectors for CONSOLE and PRINTER:



(1)	(Not connected)	
(2)-->TxD	Transmitted data	
(3)<--RxD	Received data	(IN)
(4)-->RTS	Request to send	
(5)<--CTS	Clear to send	(IN)
(6)		
(7)---GND	0V	
(8)<--DCD	Carrier detected	(IN)
(15)	(Not connected)	
(14)	(Not connected)	
(13)-->TRxC	TRxC clock	
(12)<--RI	(Not connected)	
(11)-->DTR	Data terminal ready	
(10)-> -12V	-/+12V are only for strapping and requires jumpers on the computer card. Close S4 for -12V and S3 for +12V.	
(9)-> +12V		

Serial connector on internal expansion ports, tty04 --:
I.e. with internal 4200 or 4206 boards.



(1)	(Not connected)	
(2)-->TxD	Transmitted data	
(3)<--RxD	Received data	(IN)
(4)-->RTS	Request to send	
(5)<--CTS	Clear to send	(IN)
(6)		
(7)---GND	0V	
(8)<--DCD	Carrier detected	(IN)
(15)<--TRxC	TRxC clock	(IN)
(14)<--RTxC	RTxC clock	(IN)
(13)-->TRxC	TRxC clock	
(12)<--SYNC(RI)	Ringsignal	(IN)
(11)-->DTR	Data terminal ready	
(10)	(Not connected)	
(9)	(Not connected)	

+12V and -12V is not available on these cards for jumpers.

6.1.1 Connection cables

The following terminals and modems with the corresponding cables may be connected to the system with standard cabels.

Terminal type	Cable
Facit Twist, A1400	6830
DEC VT100, Facit A2000, Facit G2000, Nokia VDU220	6884
Adm3a	6456
ABC 800/802/806	6465
Modem asynkront	6125

The following printers with the corresponding cables may be connected to the system via the printer port or any terminal port.

Facit 451x, 455x och 457x B31xx, B33xx, D2000, C5500, C7500	6464
Facit Opus2, Opus2E laserskrivare	6888
Facit 4540, 4542 (med facit interface 5145)	6889

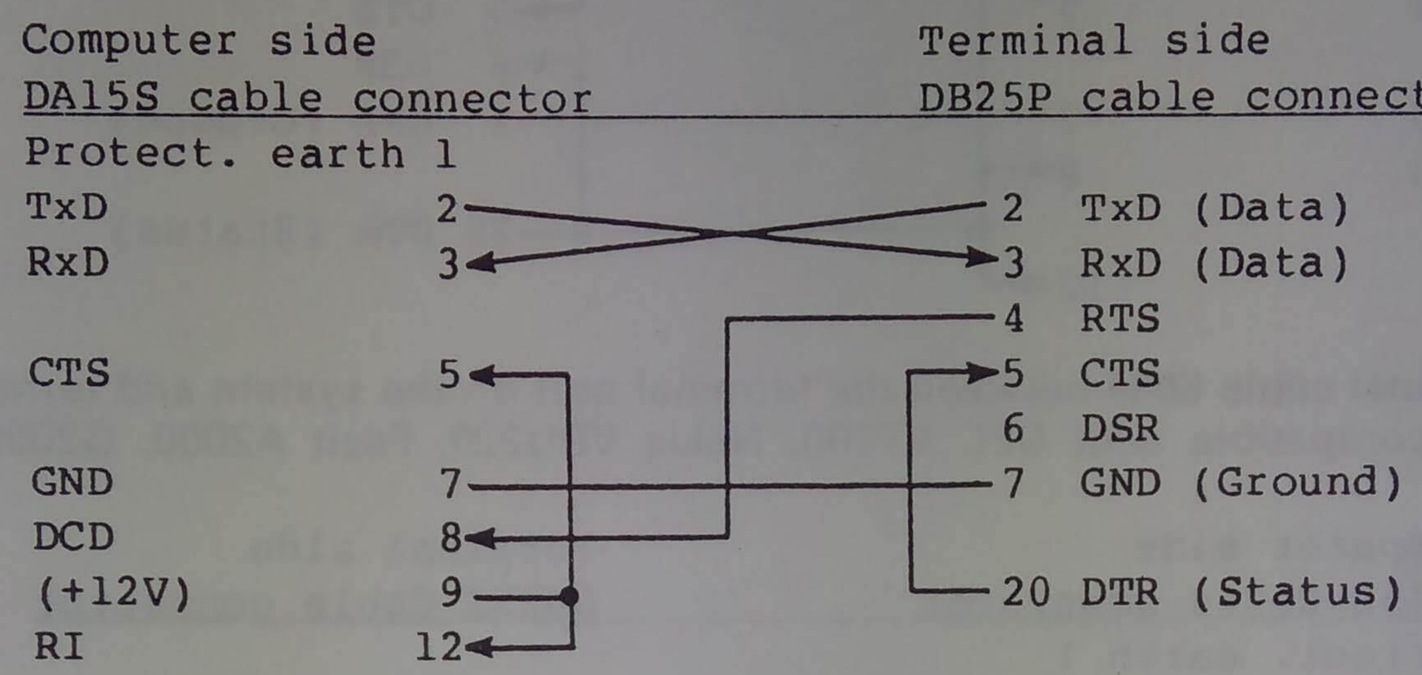
For printers with SP1 or Centronic interface a DataBoard card with associated cable is used.

Also for a number of other printers the cables above may be used, but sometimes they need to be modified if the printer gives the ready/busy signal on another connector pin than the standard. Compare the figures.

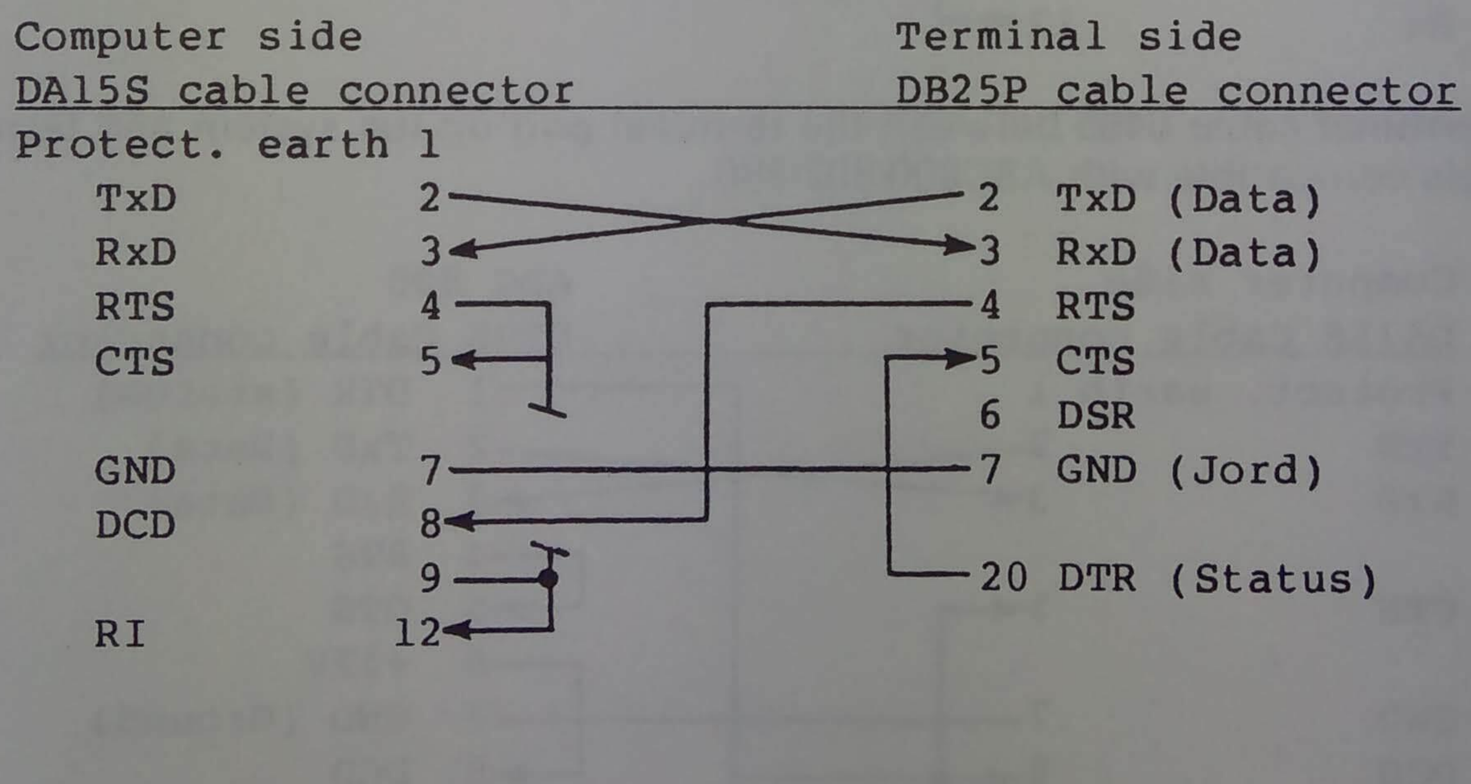
6.1.2 Connection, asynchronous terminals

* Serial asynchronous communication V.24 (RS232C)
* A 15-pin DA15S cable connector, for connection to DS90.
* The communication parameters are defined by software.
* XON/XOFF signaling from the terminal is supported

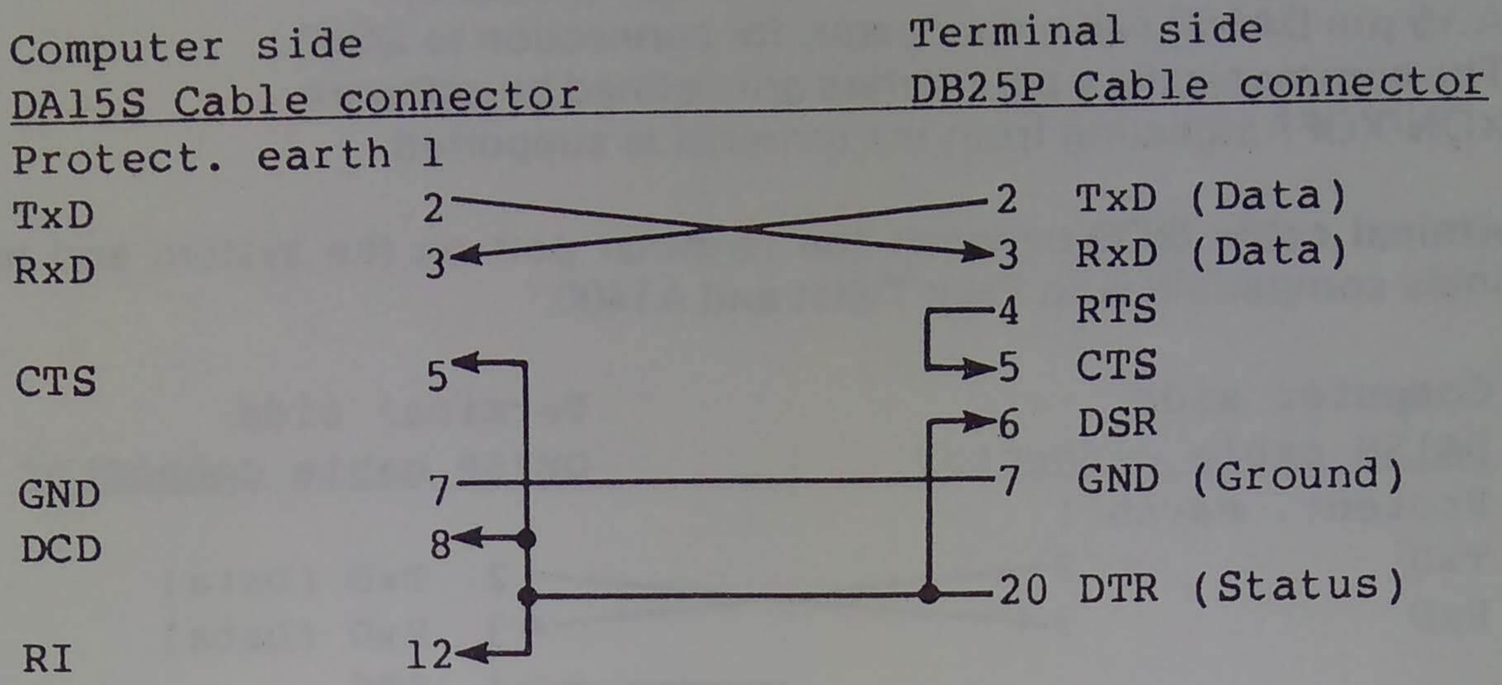
Terminal cable 6830 between the terminal port on the system and terminals compatible with Facit Twist and A1400.



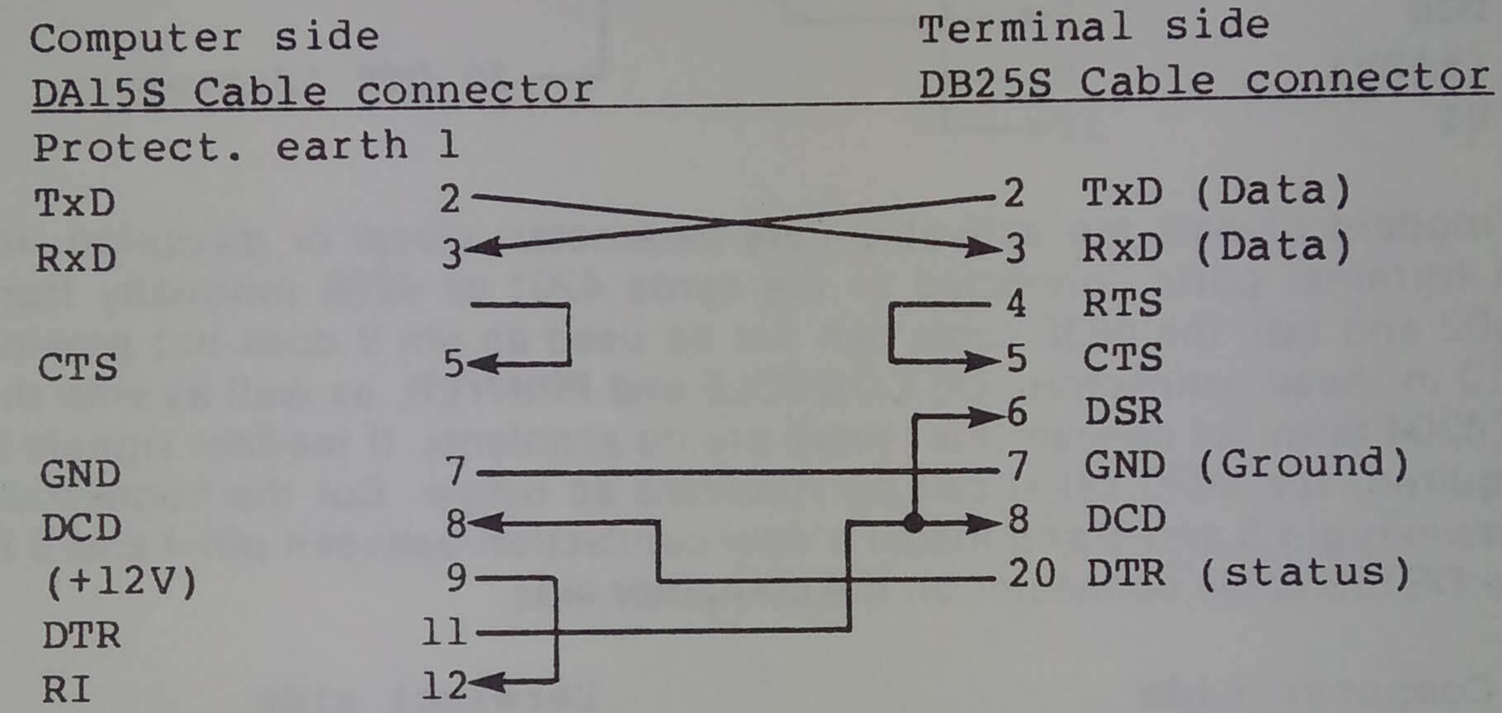
If modem signals are activated (**stty**-parameter -clocal or **cu**-option -m) on terminal ports connected to the cards 4201 or 4206 (normally from tty02 and up), the 6830 cable can not be used as pin 9 does not provide +12 in these connectors. On CONSOLE and PRINTER, as well as with the TC4204 terminal concentrator there are no problems. If modem signals is required, the 6830 cabel can be modified as below. Cut the connection between pin 9 and 5 and install a new connection between pin 4 and 5 in the DS15S cable connector on the computer side.



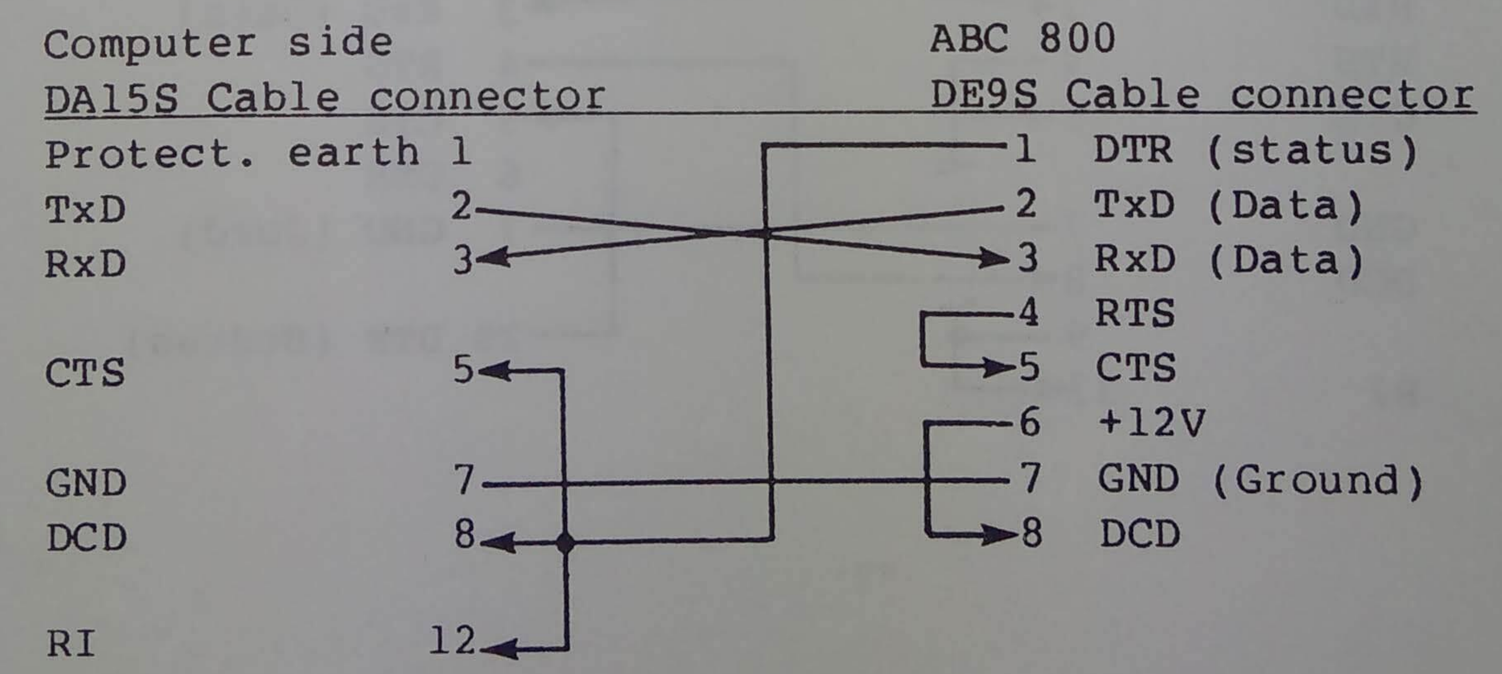
Terminal cable 6456 between the terminal port on the system and terminals compatible with ADM3A and others.



Terminal cable 6884 between the terminal port on the system and terminals compatible with DEC VT100, Nokia VDU220, Facit A2000, G2000.



Terminal cable 6465 between the terminal port on the system and terminals compatible with ABC800/802/806.



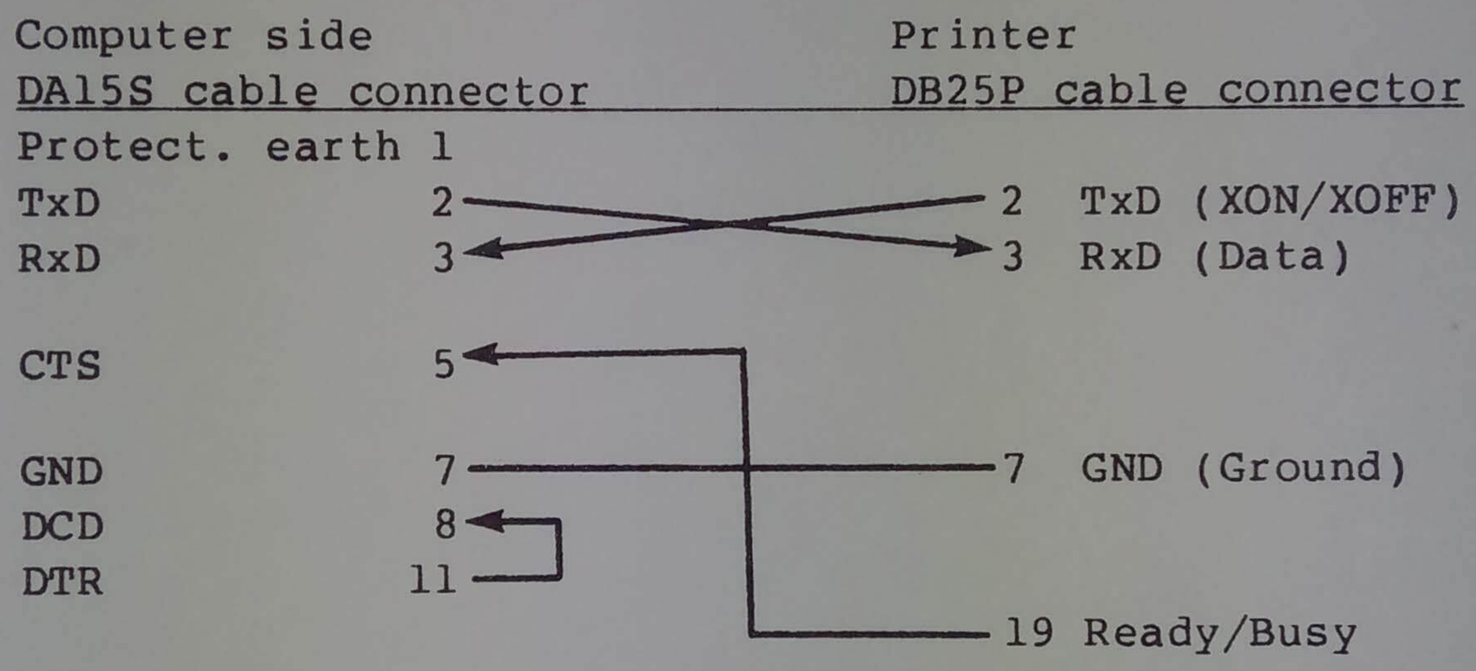
6.1.3 Connection, printers

* Serial asynchronous communication V.24 (RS232C)
* A 15-pin DA15S cable connector for connection to DS90.
* The communication parameters are defined by software.
* XON/XOFF signaling from the printer (buffer full) is supported.
  XON has the ASCII value 17 dec., DC1 (corresponds to CTRL-Q).
  XOFF has the ASCII value 19 dec., DC3 (corresponds to CTRL-S).

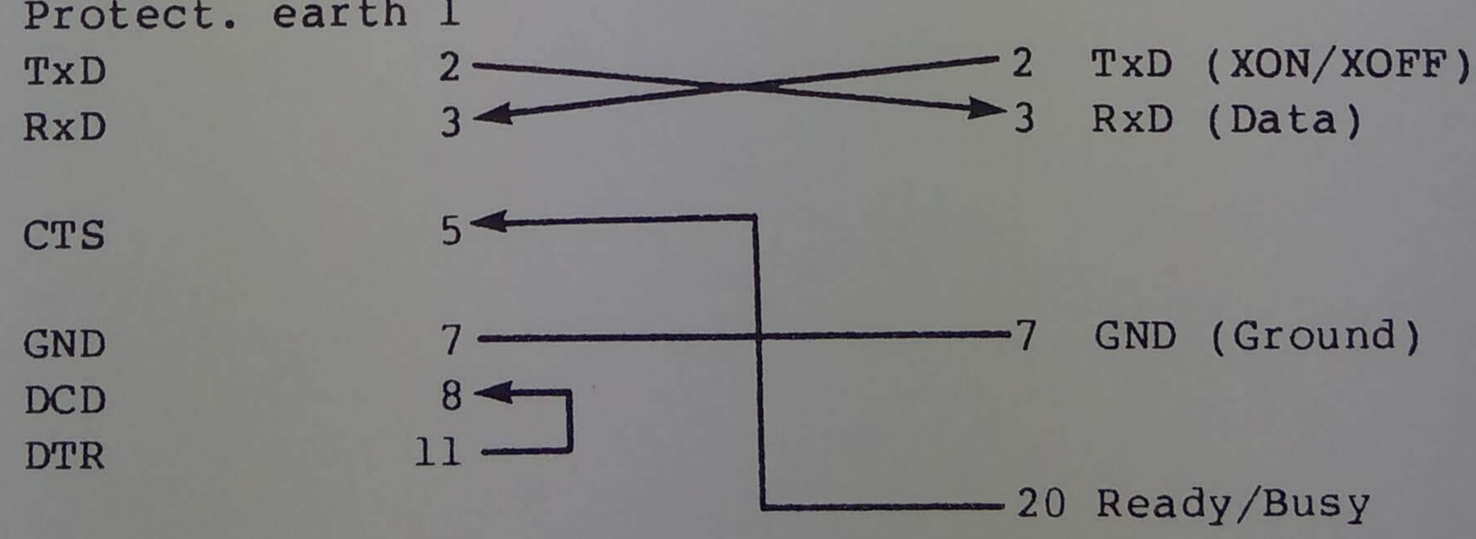
  These parameters are default values. They can be changed when needed.

For the printer port /dev/lp, a ready/busy signal from the printer is supported as a buffer full signal. The computer requires a high signal at CTS input to send characters to the printer.

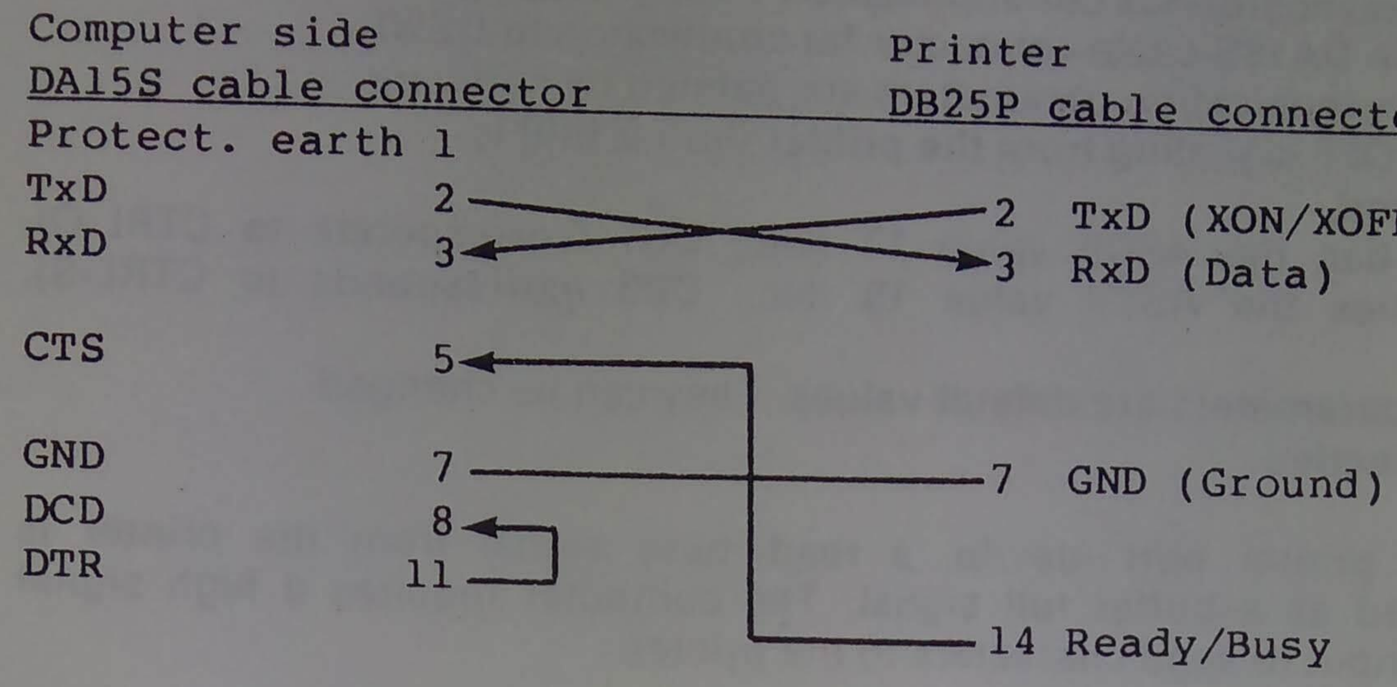
Printer cable 6464 between the printer port on the system and a printer compatible with Facit 451x, 455x, 457x, B31xx, B33xx, D2000, C55xx, C75xx.



Printer cable 6888 between the printer port on the system and a printer compatible with Facit Opus2, Opus2E.



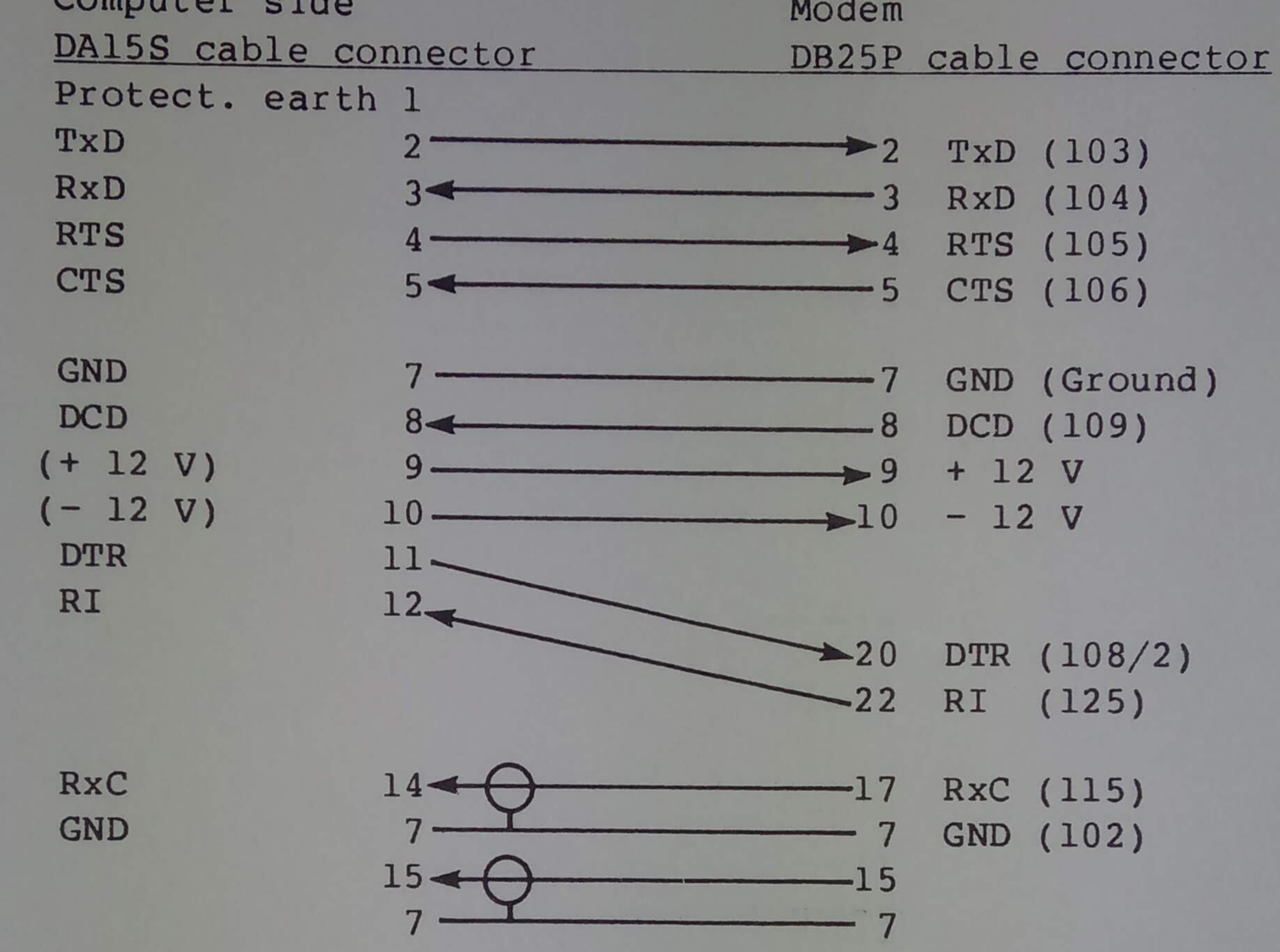
Printer cable 6889 between the printer port on the system and a printer compatible with Facit 4540 and 4542.



6.1.4 Connection, Modems

* Serial communication V.24 (RS232C)
* A 15-pin DA15S cable connector for connection to DS90.
* Only asynchronous communication is used.
* The communication parameters are defined by software.

Modem cable 6125 between a terminal port on the system and a modem.



As the cable only is used for asynchronous communication, the signal RxC is not needed. Most modems do not need + 12 V and - 12 V. In the CONSOLE and PRINTER connectors +12V/-12V are provided if the corresponding jumpers are closed on the computer board. +12V can be used for a special adapter or modem with low current consumption. On other internal expansion ports these voltages are not available. See the connectors in the beginning of chapter 6.1. If terminal concentrator boards are used, see **SA** or the TC4204 Release Notice.

A. SIV Editor

A. SIV Editor, Version 3

siv is a screen oriented editor for the system. You start **siv** by typing "siv" or "siv filename". If you use "siv filename", **siv** reads the file "filename" into the buffer when it starts.

siv may also be started with the options "-c", "-r", "-s", "-x" (you may set more than one option at the same time).

The "-c" option means that you want to continue with the file you last worked with. The "-c" option can only be used if the last filename has been saved in the file ".siv_file" (See the "-s" option).

The "-r" option states that shell commands cannot be invoked. Neiter can a new shell be started from **siv**. (Useful if you want to lock out users from shell.)

The "-s" option means that **siv** will save the current filename in the file ".siv_file" in the current directory, when **siv** is terminated in a normal way with CTRL-X CTRL-F.

The "-x" option is used when the current terminal requires XON/XOFF handshaking. The keys CTRL-S (XOFF) and CTRL-Q (XON) are then moved to CTRL-] and CTRL-\. Many terminals require handshaking with XON/XOFF when the baudrate is high.

Before starting **siv**, you have to set the shell variable TERM to your terminal type by writing "TERM=terminaltype". Then you write "export TERM". Normally, this is done automatically in your .profile.

siv works in the most favourable way at a VT100-compatible terminal. However, it may also be run on other terminal types, e.g. adm3a.

It is possible to enable **siv** to print error messages in Swedish by defining the shell variable LANGUAGE=swedish before starting **siv**.

A.1 What siv looks like on the screen.

siv uses the entire screen or the entire window in which it is invoked. The second last line holds information about the version of **siv**, the curren file name, and the line with the cursor. The last line is used for displayin error messages and for input of file names, search strings and so on.

If you use non-printable characters in a text, they are shown as "\" followed by three octal digits. These digits represent the ASCII-code for the character. Tab is printed as blank spaces up to the next tab-stop. Note that non-printable characters can **not** be entered with \ but with the command (CTRL-Q).

Rows that are too long for the screen are printed over several screenlines. At the end of each screenline, except the last one, an "\" is printed to show that the line will continue. **siv** can manage lines of up to 1024 characters. If the line is longer, **siv** divides the line into two or more lines.

A.2 Commands - Overview

The commands in **siv** consist of one or two characters. The most common commands are control characters, e.g. CTRL-F (press F at the same time you press CTRL). Some commands consist of CTRL-X or ESCAPE followed by one character, e.g. CTRL-X CTRL-S (first press CTRL-X and then CTRL-S), or ESC F (first press ESCAPE and then F).

On terminals without the ESCAPE key, you can get ESCAPE by pressing CTRL-[.

A.3 Undoing a command

If, after pressing CTRL-X or ESC, or when **siv** requests an argument, file name, a search string or a new string for some purpose, you want to have it undone, you can press CTRL-G, in which case **siv** interrupts the command and awaits a new command.

A.4 To terminate siv

siv can be terminated with two different commands:

CTRL-X CTRL-F
Saves the text in a file (see CTRL-X CTRL-S) and terminates **siv.** This is the normal way to exit from **siv.** If the file already existed, the old file is given the same name but with the extension .BAK. If **siv** was started with the -s option, the filename is entered in the file ".siv_file" in the current directory at the termination with this command. This means that **siv** can find the file at restart with the option -c. The program is then terminated.

CTRL-C
Exits without saving the contents in a file. The program asks if you really want to quit. Is that the case, you shall answer "y", otherwise "n".

CTRL-X!
Enables you to execute a SHELL command without leaving **siv**. When the command has terminated, **siv** awaits a key strike before it returns, thus giving a possibility to read a printout from the command.

CTRL-SHIFT-O (ASCII-code 31 decimal)
Will start a new shell without leaving **siv**. When the shell is terminated (with CTRL-D) **siv** continues from where it was halted. The key combination is not the same on all terminals. E.g. Facit Twist uses the key combination CTRL-SHIFT-_.

The last two commands cannot be used if **siv** was invoked with the option "-r".

A.5 Inserting of text

All printable characters (characters, digits, !"#$%&/()=?+*'_:.;,><, space) are inserted at the cursor position and eventual previous text on the line is pushed to the right.

RETURN or **LF** inserts a linefeed at the cursor position. TAB or CTRL-I inserts a TAB sign (shown as spaces). CTRL-O inserts a linefeed, but the cursor remains on the same line.

CTRL-Q
Inserts the next character at the cursor position, even if it is a control character. CTRL-Q pays no attention to if the character is redefined in ".siv".(See section A.13)

CTRL-
Is used instead of CTRL-Q if **siv** was started with the option -x.

A.6 Move commands

CTRL-B
Moves the cursor backwards one position

CTRL-F
Moves the cursor forwards one position

CTRL-P
Moves the cursor up one line

CTRL-N
Moves the cursor down one line

CTRL-V
Next screen page (Compare CTRL-Z in section A.10)

ESC-V
Previous screen page (Compare CTRL-Z in section A.10)

ESC-F
Moves the cursor to the next word

ESC-B
Moves the cursor one word backwards

CTRL-A
Moves the cursor to the beginning of the line

CTRL-E
Moves the cursor to the end of the line

ESC-<
Moves the cursor to the beginning of the text

ESC->
Moves the cursor to the end of the text

CTRL-X CTRL-X
Changes the positions for the mark and the cursor (compare with CTRL-@)

On terminals that are VT100-compatible you may also use the arrow keys (cursor keys) to move around in the text.

A.7 Deletion of text

CTRL-D
Deletes the character in the position of the cursor.

CTRL-H (or BS =backspace)
Deletes the character before the cursor.

CTRL-K
If you are at the end of the line, the linefeed is deleted and put in the delete buffer, i.e. the line is merged with the next line. If you are not at the end of the line, all text from the cursor to the end of the line is deleted and placed in the delete buffer. If you perform several CTRL-K in sequence, the text is appended to the delete buffer. If you perform another command, the delete buffer will be emptied before the next CTRL-K.

CTRL-W
Deletes all text between the mark and the cursor (compare with CTRL-@), and puts the text in the delete buffer. (The delete buffer will first be emptied).

CTRL-@ (ASCII-code decimal 0)
Puts a mark before the cursor. The mark will remain in its position until it is put somewhere else. When **siv** is invoked there is no mark. Compare with CTRL-W and CTRL-X CTRL-X. Some terminals can not give CTRL-@ as this character has the ASCII value 0. E.g. on Facit Twist the key combination CTRL-space is used instead.

CTRL-Y
Inserts the text from the delete buffer before the cursor position. The contents of the delete buffer is not affected, which means that with CTRL-Y you may undo earlier deletion if you do not move the cursor.

ESC-D
Deletes the word after the cursor. Characters that are interpreted as word separators are defined in the file ".siv" (See section A.13).

A.8 File handling commands

CTRL-X CTRL-S
Saves the text in the file stated on the information line. If no file is stated **siv** will ask for a file name. If the file existed earlier the old file is saved by having the same name but with the extension .BAK.

CTRL-X CTRL-]
Is used instead of CTRL-X CTRL-S if **siv** was started with the -x option.

CTRL-X CTRL-F
This is the normal way to terminate **siv** and the text is automatically saved in the same way as with CTRL-X CTRL-S before the termination. See section A.4 for more details.

CTRL-X CTRL-W
Saves the text in a named file.

CTRL-X CTRL-V
Reads a file. The contents of the buffer are destroyed.

CTRL-X CTRL-R
Same as CTRL-X CTRL-V.

CTRL-X CTRL-I
Inserts the contents of a file between the line where the cursor is and the previous line.

CTRL-X CTRL-D
Asks for a new working directory and moves on to it. If you want to see which directory the current one is, give "." as the new working directory.

CTRL-X CTRL-T
Saves the text on a file exactly as it looks on the screen, i.e. control characters are printed as "\nnn" where nnn is equal to three octal digits.

CTRL-X CTRL-L
Loads the file ".siv" from your home directory.

A.9 Commands for searching and changing strings

CTRL-S
Searches forward for the next occurance of a text string. **siv** asks for a text string (entered with RETURN), and goes to the next occurance of it. If the text string does not exist after the cursor, **siv** will let you know.

CTRL-]
Is used instead of CTRL-S, if **siv** was started with the -x option.

CTRL-R
Searches backwards in the same way as CTRL-S.

If you press CTRL-S CTRL-S (CTRL-] CTRL-]), **siv** will search for the previous string again. The same with CTRL-R CTRL-R. This means that if you do not find the string with CTRL-S CTRL-S you may try CTRL-R CTRL-R if you believe the string is somewhere before the cursor.

ESC-Q
Replaces all occurences of a specified string with a new string. The program will ask before each replacement. First **siv** asks for the old string and then the new string. Then **siv** searches for the first occurence of the string. Now you may choose between the following three commands:

SPACE
Replace this string and search for the next.

N or CTRL-H (or BS)
Do not replace this string. Search for the next.

!
Replace all occurances.

CTRL-G
Interrupt the command.

If you do not enter a string to replace (giving only RETURN), **siv** will use the previous search string. The string, previously used for replacement, will be used as search string the next time you use CTRL-S CTRL-S (CTRL-] CTRL-]) or CTRL-R CTRL-R.

A.10 Commands to rewrite the screen

CTRL-L
Rewrites the screen. The line at which the cursor is positioned is centered. Also rewrites the information line and deletes error messages.

CTRL-Z
Moves the text and the cursor up one line on the screen. The command has no effect if the cursor already is on the first line on the screen.

A.11 Arguments to commands

CTRL-U
Is used to perform a command several times. **siv** then asks for an argument. If no argument is given default is 4. You may enter any number lower than 32000. You may also use CTRL-U in the argument. This means that the previous argument will be multiplied by 4. CTRL-U followed by digits means that the previous argument will be cleared and you start from the beginning. Example:

```
CTRL-U 42 ESC-B           Will move backwards 42
                          words.
CTRL-U CTRL-U *           Inserts 16 asterisks in
                          the text.
CTRL-U 33 CTRL-U CTRL-B   Moves backwards 33*4=132
                          characters.
CTRL-U 5 CTRL-U 7 ?       Inserts 7 question
                          marks.
```

A.12 Other commands

CTRL-T
Lets the two characters preceding the cursor change places.

A.13 The file ".siv"

The file ".siv" is a control file for **siv**. It is read each time **siv** is invoked. The file contains definitions for word separators, macros, and ordinary characters. If ".siv" does not exist in the home directory of the user, **siv** tries to read "/etc/.siv".

If "/etc/.siv" does not exist, **siv** creates a standard ".siv" in the user's home directory. In this file, all characters, except letters, digits, and underscore "_" are defined as word separators. All characters are defined in a normal way. The following macros are defined as standard:

ESC-W Is defined as CTRL-W CTRL-Y. In other words, copies the text between the mark and the cursor and puts it in the delete buffer

ESC-w Is defined as above, see ESC-W.
ESC-l Prints out a text string.
ESC-L Prints out a text string that contains two two lines. The text strings are only examples to show how a command is defined.

ESC-[-A Is defined as ESC-O-A
ESC-[-B Is defined as ESC-O-B
ESC-[-C Is defined as ESC-O-C
ESC-[-D Is defined as ESC-O-D

The format of the file ".siv" is as follows:

The first line contains the word separators. This line is ended with a NULL-character (ASCII value 0) The control characters are always interpreted as word separators and therefore they need not be specified on this line.

Then follows 32 macro definitions with the following format:

```
x\377yyyy\377
```

Where x is a character and yyyy are the characters that are sent to **siv** when you give the command ESC-x.

yyyy can be up to 61 characters long. They can be **siv**-commands, text or a combination of these. For example, the definition

```
*\377\020\001/*\005*/\377
```

performs the following when you press ESC-*:

Moves up one line (CTRL-P), goes to the beginning of the line (CTRL-A), inserts "/*", goes to the end of the line (CTRL-E) and inserts "*/".

The **siv** commands in a macro definition are always represented by their ASCII values in octal notation.

Example:

```
CTRL-A = \001
CTRL-B = \002
etc.
```

Note that control characters are inserted in the file with CTRL-Q (or CTRL-\ if the x-option is given) and not by writing e.g "\020".

A macro definition may also extend over several lines:

```
L\377Free text
two lines
\377
```

If you happen to delete a "\377" -character by mistake while making changes in ".siv" , you can replace it by copying one of the other "\377" to the delete buffer with the command CTRL-@ CTRL-F CTRL-W CTRL-Y. After that, you copy from the buffer with command CTRL-Y.

After the macro definitions follow 128 lines with the translation table for all ASCII characters typed on the keyboard. Here you may redefine the characters if you like. For example, you might want to redefine DELETE as BS.

A line contains the following; first the character that comes from the terminal, then a TAB - character, then the characters function. For example:

```
\010            \020
```

This means that BS will be interpreted as CTRL-B. The lines that contain the definitions for TAB and LF look a bit strange. Especially the definition for LF that takes three lines!

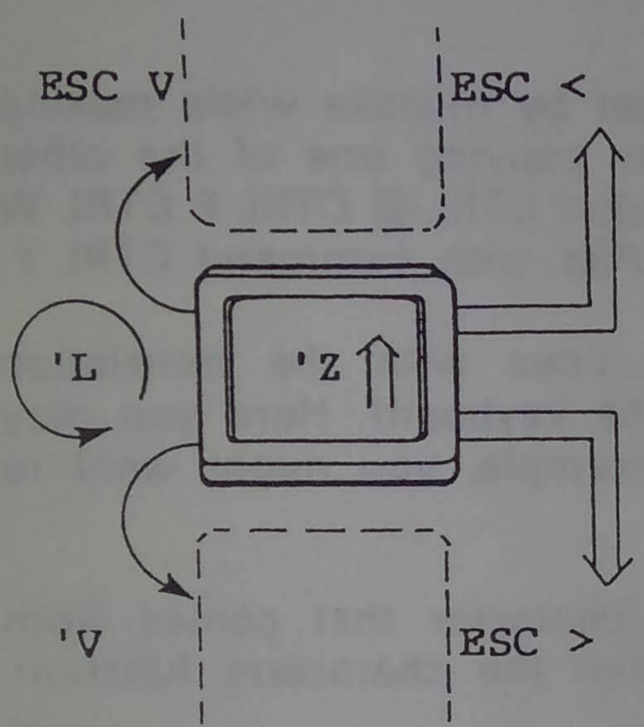
If you by mistake should make some crazy definitions in ".siv", and you cannot get back the original definitions when you use **siv**, then you can do the following: Rename ".siv" to ".siv-". Then invoke **siv**, make the changes in ".siv-" and save it again as ".siv".

A.14 Summary of commands

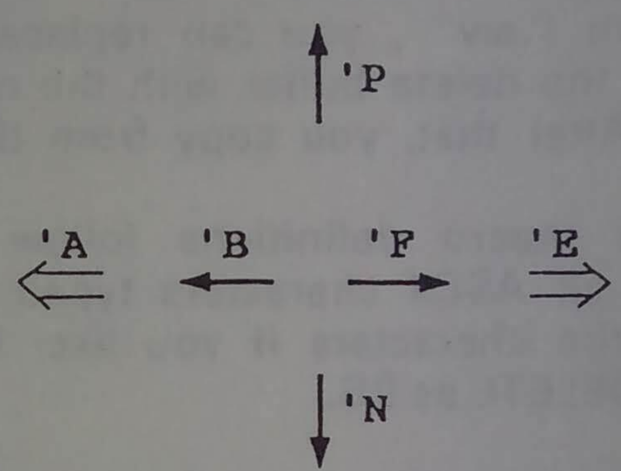
In this summary control characters have been abbreviated. CTRL-X is shown as 'X. NOTE: This summary does not include all the commands.

Function	Command
Start of siv	siv filename
Abort siv	'C
Read a new file	'X 'V (or 'X 'R)
Write file, exit from siv	'X 'F
Write to new file	'X 'W
Execute shell command	'X !command
Change directory	'X 'D
Repeat siv command	'U nn command (default nn=4)

Screen commands



Cursor commands



Erase	Command
Character left	'H
Character under cursor	'D
Word backwards	ESC D
Line forward	'K
Block	'@ (cursor to end of block) 'W
Insert block	'Y

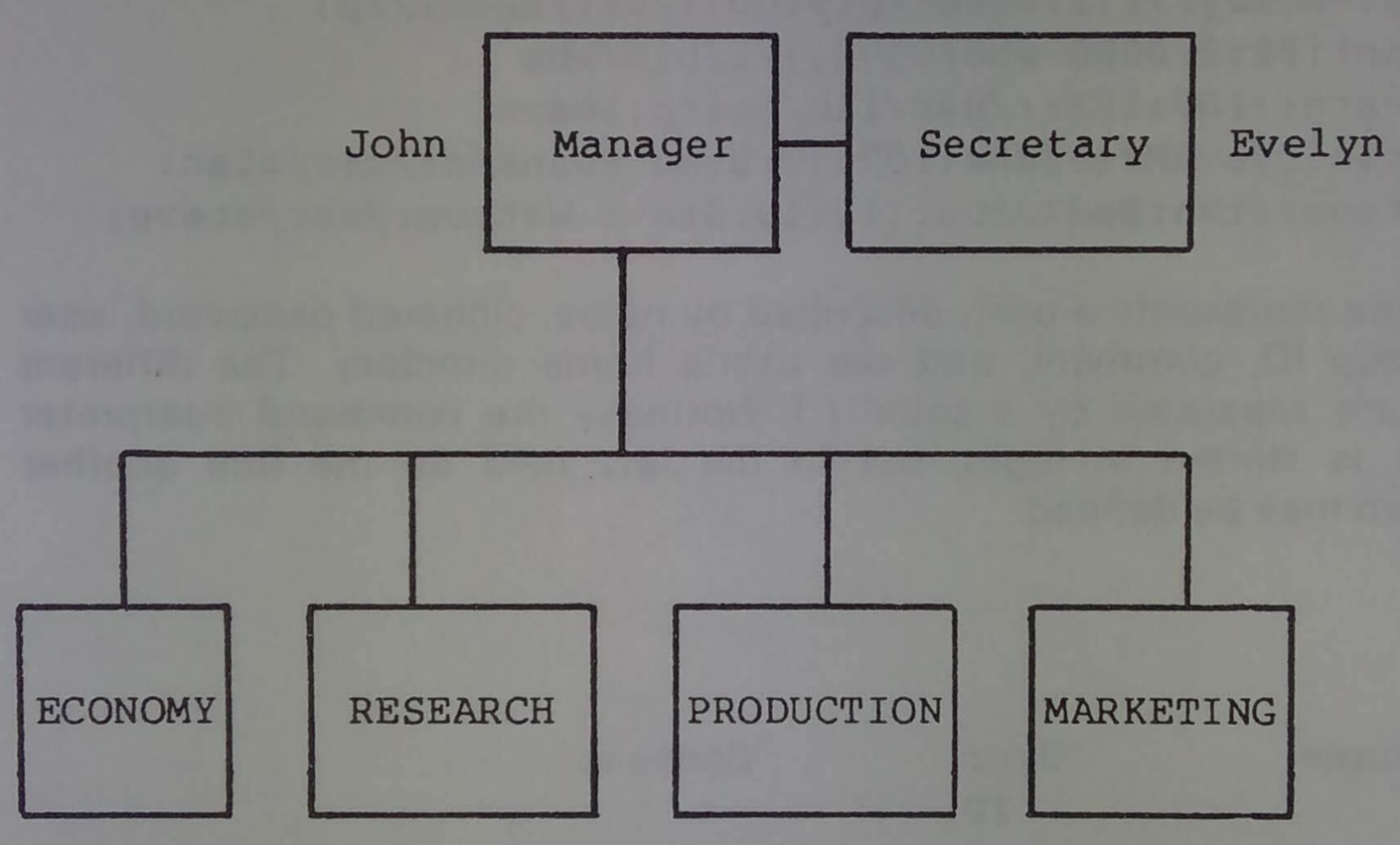
Search and replace	Command
Search forward	'S (or '])
Search backward	'R
Replace string	ESC Q

B. Example of how to arrange a system

When a system is arranged in a company there are many factors to keep in mind. How shall the different groups look like, how to set user numbers, etc. All this should be considered before the system is taken into use. It is important that the system is correctly arranged from the start. If not, situations may occur where it is difficult to change the build-up.

Below we want to give an example on how to organize a system that functions well in most cases.

The head of the company is the managing director. Under him there are four different departments: economy, research, production, and marketing. This company has one computer system that shall be used by all four departments and the manager. The organisation may look like this:



The file /etc/passwd may look as follows when two ordinary users have been added. (Stan and Steve):

Example on /etc/passwd file:

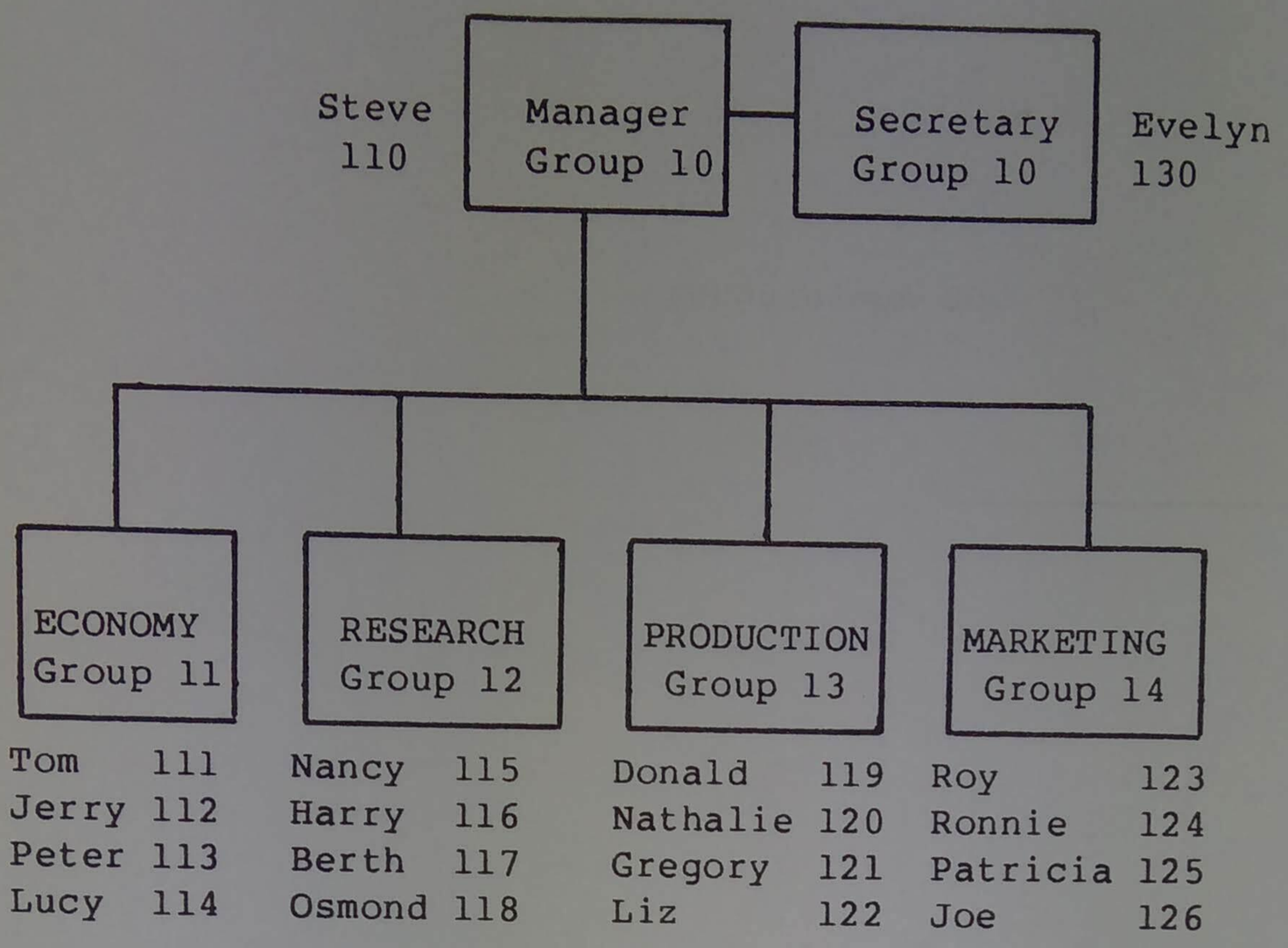
```
root::0:3:0000-Admin(0000):/:
daemon:*noway*:1:12:0000-Admin(0000):/:
bin:*noway*:2:2:0000-Admin(0000):/bin:
sys:*noway*:3:3:0000-Admin(0000):/usr/src:
adm:*noway*:4:4:0000-Admin(0000):/usr/adm:
uucp:*noway*:5:1:0000-uucp(0000):/usr/lib/uucp:
nuucp::6:1:0000-uucp(0000):/usr/spool/uucppublic:
        /usr/lib/uucp/uucico
sync::20:1:0000-Admin(0000):/:/bin/sync
rje:*noway:68:8:0000-rje(0000):/usr/rje:
shqer:*noway:69:8:0000-rje(0000):/usr/rje:
lp:*noway:71:2:0000-lp(0000):/usr/spool/lp:
who::72:2:0000-who(0000):/:/bin/who
learn::102:102::/usr/lib/learn:learn
stan:3Uuv4MJme0Khk:109:10:Stan Swanson:/usr/stan:
steve:rrFbrSmdXJo0s:110:10:Steve Watson:/usr/steve:
```

Each line represents a user, described by name, ciphered password, user ID, group ID, comment, and the user's home directory. The different fields are separated by a colon (:). Normally the command interpreter /bin/sh is started at login, but in the last field on the line another program may be defined.

```
User name          User        Comment
                     ID
  ▲                  ▲           ▲
steve:rrFbrSmdXJo0s:110:10:Steve Watson:/usr/steve:
        ▼            ▼                    ▼
    Ciphered      Group ID          Home directory
    password
```

When setting up a system the user ID, the group ID, and the home directory are the most important items. In the above described company it is natural that each department has its own group ID. As user IDs it is preferable to choose the employment number or another combination of digits unique for each person.

When selecting group IDs it is preferable to start the numbering from 10 as lower numbers already are occupied by sys, bin and others. On the next page is shown how our company might organize their system.



The managing director in our company has the user name steve, user ID 110, and group ID 10. His entry in the file /etc/passwd may look as follows:

```
steve:rrFbrSmdXJo0s:110:10:Steve Watson:/usr/steve:
```

The file- and directory structure of the system is described in chapter 5, with information on how groups of users can use the same files. See also UM.

INDEX

A

B

C

D

E

M

N

O

P

R

S

T

U

V

W

Evaluation of documentation:

Installation manual DS90-00 089-9703-00(A)

We are very much interested in your impression of our documentation and in any suggestions you might have for its improvement. We would be very greatful if you would take a few minutes of your time to answer the questions below and mail this page back to us.

1. Is it easy to find the information you need? Is there any information you need that you can not find?

---

2. Is the text easy to read and easy to understand? Please cite page and paragraph of difficult parts.

---

3. Is the documentation logically organized? What changes, if any, do you suggest?

---

4. Are there enough examples, and do they describe the relevant parts?

---

5. How can the documentation be improved? Please cite specific examples.

Name:_____ Company: _____
(Optional information)

Our address is pre-printed on the backside of this form. Fold it, staple, and it is ready to send by mail.

Thank you for your help!

Diab Data AB
Documentation
PO Box 2029
S-183 02 TÄBY
Sweden

NAME

intro - introduction to system calls and error numbers

SYNOPSIS

#include <errno.h>

DESCRIPTION

This section describes all of the system calls. Most of these calls have one or more error returns. An error condition is indicated by an otherwise impossible returned value. This is almost always -1; the individual descriptions specify the details. An error number is also made available in the external variable errno. Errno is not cleared on successful calls, so it should be tested only after an error has been indicated.

Each system call description attempts to list all possible error numbers. The following is a complete list of the error numbers and their names as defined in <errno.h>.

1 EPERM Not owner
Typically this error indicates an attempt to modify a file in some way forbidden except to its owner or super-user. It is also returned for attempts by ordinary users to do things allowed only to the super-user.

2 ENOENT No such file or directory
This error occurs when a file name is specified and the file should exist but doesn't, or when one of the directories in a path name does not exist.

3 ESRCH No such process
No process can be found corresponding to that specified by pid in kill or ptrace.

4 EINTR Interrupted system call
An asynchronous signal (such as interrupt or quit), which the user has elected to catch, occurred during a system call. If execution is resumed after processing the signal, it will appear as if the interrupted system call returned this error condition.

5 EIO I/O error
Some physical I/O error has occurred. This error may in some cases occur on a call following the one to which it actually applies.

6 ENXIO No such device or address
I/O on a special file refers to a subdevice which does not exist, or beyond the limits of the device. It may also occur when, for example, a tape drive is not on-line or no disk pack is loaded on a drive.

7 E2BIG Arg list too long
An argument list longer than 5,120 bytes is presented to a member of the exec family.

8 ENOEXEC Exec format error
A request is made to execute a file which, although it has the appropriate permissions, does not start with a valid magic number [see a.out(4)].

9 EBADF Bad file number
Either a file descriptor refers to no open file, or a read (respectively, write) request is made to a file which is open only for writing (respectively, reading).

10 ECHILD No child processes
A wait was executed by a process that had no existing or unwaited-for child processes.

11 EAGAIN No more processes
A fork failed because the system's process table is full or the user is not allowed to create any more processes.

12 ENOMEM Not enough space
During an exec, brk, or sbrk, a program asks for more space than the system is able to supply. This is not a temporary condition; the maximum space size is a system parameter. The error may also occur if the arrangement of text, data, and stack segments requires too many segmentation registers, or if there is not enough swap space during a fork.

13 EACCES Permission denied
An attempt was made to access a file in a way forbidden by the protection system.

14 EFAULT Bad address
The system encountered a hardware fault in attempting to use an argument of a system call.

15 ENOTBLK Block device required
A non-block file was mentioned where a block device was required, e.g., in mount.

16 EBUSY Device or resource busy
An attempt was made to mount a device that was already

mounted or an attempt was made to dismount a device on which there is an active file (open file, current directory, mounted-on file, active text segment). It will also occur if an attempt is made to enable accounting when it is already enabled. The device or resource is currently unavailable.

17 EEXIST File exists
An existing file was mentioned in an inappropriate context, e.g., link.

18 EXDEV Cross-device link
A link to a file on another device was attempted.

19 ENODEV No such device
An attempt was made to apply an inappropriate system call to a device; e.g., read a write-only device.

20 ENOTDIR Not a directory
A non-directory was specified where a directory is required, for example in a path prefix or as an argument to chdir(2).

21 EISDIR Is a directory
An attempt was made to write on a directory.

22 EINVAL Invalid argument
Some invalid argument (e.g., dismounting a non-mounted device; mentioning an undefined signal in signal, or kill; reading or writing a file for which lseek has generated a negative pointer) was attempted. The math functions described in the (3M) entries of this manual causes the invalid argument to be set.

23 ENFILE File table overflow
The system file table is full, and temporarily no more opens can be accepted.

24 EMFILE Too many open files
No process may have more than 20 file descriptors open at a time. When a record lock is being created with fcntl, there are too many files with record locks on them.

25 ENOTTY Not a character device
An attempt was made to ioctl(2) a file that is not a special character device.

26 ETXTBSY Text file busy
An attempt was made to execute a pure-procedure program that is currently open for writing. Also an attempt to open for writing a pure-procedure program that is being executed.

27 EFBIG File too large
The size of a file exceeded the maximum file size (1,082,201,088 bytes) or ULIMIT; see ulimit(2).

28 ENOSPC No space left on device
During a write to an ordinary file, there is no free space left on the device. In fcntl, the setting or removing of record locks on a file cannot be accomplished because there are no more record entries left on the system.

29 ESPIPE Illegal seek
An lseek was issued to a pipe.

30 EROFS Read-only file system
An attempt to modify a file or directory was made on a device mounted read-only.

31 EMLINK Too many links
An attempt to make more than the maximum number of links (1000) to a file.

32 EPIPE Broken pipe
A write on a pipe for which there is no process to read the data. This condition normally generates a signal; the error is returned if the signal is ignored.

33 EDOM Math argument
The argument of a function in the math package (3M) is out of the domain of the function.

34 ERANGE Result too large
The value of a function in the math package (3M) is not representable within machine precision.

35 ENOMSG No message of desired type
An attempt was made to receive a message of a type that does not exist on the specified message queue; see msgop(2).

36 EIDRM Identifier Removed
This error is returned to processes that resume execution due to the removal of an identifier from the file system's name space [see msgctl(2), semctl(2), and shmctl(2)].

45 EDEADLK Deadlock
A deadlock situation was detected and avoided.

DEFINITIONS

Process ID
Each active process in the system is uniquely identified by a positive integer called a process ID. The range of this ID is from 1 to 30,000.

Parent Process ID
A new process is created by a currently active process; see fork(2). The parent process ID of a process is the process ID of its creator.

Process Group ID
Each active process is a member of a process group that is identified by a positive integer called the process group ID. This ID is the process ID of the group leader. This grouping permits the signaling of related processes; see kill(2).

Tty Group ID
Each active process can be a member of a terminal group that is identified by a positive integer called the tty group ID. This grouping is used to terminate a group of related processes upon termination of one of the processes in the group; see exit(2) and signal(2).

Real User ID and Real Group ID
Each user allowed on the system is identified by a positive integer called a real user ID.

Each user is also a member of a group. The group is identified by a positive integer called the real group ID.

An active process has a real user ID and real group ID that are set to the real user ID and real group ID, respectively, of the user responsible for the creation of the process.

Effective User ID and Effective Group ID
An active process has an effective user ID and an effective group ID that are used to determine file access permissions (see below). The effective user ID and effective group ID are equal to the process's real user ID and real group ID respectively, unless the process or one of its ancestors evolved from a file that had the set-user-ID bit or set-group ID bit set; see exec(2).

Super-user
A process is recognized as a super-user process and is granted special privileges if its effective user ID is 0.

Special Processes
The processes with a process ID of 0 and a process ID of 1

are special processes and are referred to as proc0 and proc1.

Proc0 is the scheduler. Proc1 is the initialization process (init). Proc1 is the ancestor of every other process in the system and is used to control the process structure.

File Descriptor

A file descriptor is a small integer used to do I/O on a file. The value of a file descriptor is from 0 to 19. A process may have no more than 20 file descriptors (0-19) open simultaneously. A file descriptor is returned by system calls such as open(2), or pipe(2). The file descriptor is used as an argument by calls such as read(2), write(2), ioctl(2), and close(2).

File Name

Names consisting of 1 to 14 characters may be used to name an ordinary file, special file, or directory.

These characters may be selected from the set of all character values excluding \0 (null) and the ASCII code for / (slash).

Note that it is generally unwise to use *, ?, [, or] as part of file names because of the special meaning attached to these characters by the shell. See sh(1). Although permitted, it is advisable to avoid the use of unprintable characters in file names.

Path Name and Path Prefix

A path name is a null-terminated character string starting with an optional slash (/), followed by zero or more directory names separated by slashes; optionally followed by a file name.

More precisely, a path name is a null-terminated character string constructed as follows:

<path-name>::=<file-name>|<path-prefix><file-name>|/
<path-prefix>::=<rtprefix>|/<rtprefix>
<rtprefix>::=<dirname>/|<rtprefix><dirname>/

where <file-name> is a string of 1 to 14 characters other than the ASCII slash and null, and <dirname> is a string of 1 to 14 characters (other than the ASCII slash and null) that names a directory.

If a path name begins with a slash, the path search begins at the root directory. Otherwise, the search begins from the current working directory.

A slash by itself names the root directory.