

Unirex

Introductory Guide

Dansk Data Elektronik A/S  
27 July 1983

Author: Claus Tøndering

Copyright 1983  
Dansk Data Elektronik A/S



Table of Contents.

	<u>Page</u>
1. Introduction	1.1
2. What is Unirex?	2.1
3. Getting Started	3.1
3.1. Logging On	3.1
3.2. Typing to the Terminal	3.2
3.3. The Edit Operation	3.5
4. Iounits	4.1
4.1. Terminals	4.1
4.2. Printers	4.2
4.3. Disks	4.2
4.4. Files	4.3
4.5. Boxes	4.4
4.6. The Null Device	4.4
4.7. Current Unit Prefix	4.5
5. Access Rights	5.1
5.1. User Numbers	5.1
5.2. Protection Bits	5.2
5.3. Privileged Users	5.3
6. The Computer	6.1
6.1. System Crash	6.1

## 1. Introduction

This manual describes the basic features of the Unirex operating system. The manual is intended for the first time users of Unirex.

Dansk Data Elektronik A/S reserves the right to change the specifications in this manual without warning. Dansk Data Elektronik A/S is not responsible for the effects of typographical errors or other inaccuracies in this manual and cannot be held liable for the effects of the implementation and use of the structures described herein.



## 2. What is Unirex?

Unirex is the operating system of the Unimax computer. Unirex is the program that controls the execution of all other programs on the computer.

The user does not see Unirex directly. The user merely communicates with one or more running programs such as a word processing system, a BASIC interpreter, a bookkeeping package, etc.; but behind it all the omnipresent Unirex is controlling and checking the user's programs.

Thus the user is normally not aware of the presence of Unirex, but, nonetheless, the knowlegde of a few basics is required in order to use the system properly. Also, a general knowlegde of what is going on inside the computer will help the user to make better use of the system.



### 3. Getting Started.

In the following section reference will often be made to the System Administrator. This is the person (or persons) in your company who is responsible for the Unimax computer.

Before you can do anything on the computer, you need to be allowed access to the computer. Contact your System Administrator. He or she will assign you

- 1) a 'user name',
- 2) a secret 'password',
- 3) a 'user number', and
- 4) an 'operator service program'.

The user name may simply be your first or last name. The password may be any combination of letters and numbers you desire. The user number will be discussed in detail later. The operator service program is the program that you will be running on the computer once you get in contact with it; more about this later.

#### 3.1. Logging On.

Let us assume that you have access to a running Unimax computer. Connected to the computer are a number of terminals. Find a vacant terminal. When nobody uses the terminal, the screen will normally show some introductory picture like the following:

```
*****
*           Welcome to Unirex version 23.06.1983           *
*                                                         *
*                                     Terminal number 4      *
*****
```

This picture is called the logon picture. Its actual layout of this picture may vary. It may be redesigned by the System Administrator.

Have a look at the keyboard. The keyboard consists of the keys found on normal typewriters, plus a set of so-called function keys. The actual layout of the keyboard will depend on what kind of terminal you are using, so if the following description confuses you, talk with your System Administrator.



One key is the 'attention key'. It will normally be labelled 'ESCAPE' or 'ESC'. This key is used to 'kick' the system, that is, change the flow of events inside the computer. If you press this key, the text 'Enter user name:' will appear on the screen. Type your user name and press the RETURN key. Now the text 'Enter password:' will appear, and you must type your secret password followed by the RETURN key. Note that the password will not be displayed on the screen as you type it. Because of this, somebody watching over your shoulder will have a hard time discovering your password.

If you made a mistake when typing your user name or password, the text 'Rejected' will be displayed, and you will have to try to enter your user name again.

If you have typed everything correctly, the text 'You are now logging on to processor number: 1' will appear. The number may not be '1'. This will be discussed later. For the present, simply press the RETURN key.

After a few seconds a program will start running. This is your 'operator service program' assigned to you by the System Administrator. It may be some specialized program package, such as a word processing program or a bookkeeping program, or it may be a general operator communication program, such as 'vox', which is described in the 'Unirex Utility Guide' manual. What you must do hereafter depends on what operator service program the computer is running for you.

What you have just done is termed 'logging on'. When you stop the execution of the operator service program, the system will return to the initial 'Welcome to Unirex display'. This is termed 'logging off'.

### 3.2. Typing to the Terminal.

You have already discovered a few features about the terminal you are working with. First, you will have noticed the block or underscore that moves in front of your text as you type, always indicating the place in which the next character will go. This indicator is called the 'cursor'. Second, you will have noticed that, as you entered your user name, the keys that you pressed were shown on the screen, whereas this was not the case when you entered your password. Normally the terminal will display what you type; we say that the computer 'echoes' your keystrokes.



Terminals differ from normal typewriters in one very important respect: On a typewriter the letter l (small L) is often used for the digit 1 (one) and the letter O (capital o) is often used for the digit 0 (zero). This is not the case on a terminal. The terminal is equipped with keys bearing the digits zero and one, and these keys must be used when you want to type the digits, as most programs will not accept the letters l and O. Some terminals and printers display the digit zero with a slash through it (thus: 0) in order to distinguish between the digit and the letter. (This, of course, does not help Norwegians and Danes, for whom 0 is a letter in its own right.)

As you type a line to the terminal, you may make mistakes. In most cases the line you are writing will not be sent to the computer program before you press the RETURN key. This means that as long as you do not hit the RETURN key you are allowed to change the contents of the line. A number of keys on the terminal aid you in editing the line, these keys will be described in the following paragraphs. You need not worry about committing everything below to memory, merely read the following, and return to it as you become acquainted with the computer.

The left arrow and right arrow keys move the cursor left or right. You may use these keys to position the cursor on an erroneous letter and then hit the correct key. The new letter will replace the old one.

The start-of-line and end-of-line keys (on some terminals labelled with a double left and double right arrow, respectively) will move the cursor to the beginning of the line and the end of the line, respectively. The end of the line is the position following the last visible character on the line. On many terminals you have to hold down the SHIFT key in order to activate the end-of-line function.

The line you are writing has a maximum length. If you try to type more characters than the length of the line, you will hear a beep, and the cursor will not move beyond the end of the line.

The insert character key makes space for a new character at the cursor position. The rightmost character will be lost, if the line is full. Suppose, for example, that your line looks like this (the underscore represents the cursor):

Friends, Romans, countrymn, lend me your ears.  
Pressing the insert character key will change the line to:



Friends, Romans, countrymn, lend me your ears.  
Now you may enter the missing letter.

The delete character key deletes the character at the cursor position. Suppose, for example, that your line looks like this (the underscore represents the cursor):

Friends, Romans, countrymean, lend me your ears.  
Pressing the delete character key will change the line to:

Friends, Romans, countrymen, lend me your ears.

The erase to end-of-line key deletes the characters at and to the right of the cursor. Suppose, for example, that your line looks like this (the underscore represents the cursor):

Friends, Romans, countrymean, lend me your ears.  
Pressing the erase to end-of-line key will change the line to:

Friends, Romans, countryme\_

The erase line key erases the whole line and moves the cursor to the start of the line.

The tab key moves the cursor to the next tab stop. Tab stops are located at every eighth position on the line.

The underscore key (\_) underlines the character at the cursor position. On some terminals an underlined character will be shown black on white instead of underlined. If you type a new character on top of an underlined character, the underline will disappear.

The rub key (labelled rubout, del, or delete on some terminals) removes the underline from an underlined character.

Most other keys on the keyboard will terminate the line editing as the RETURN key does it.

Special attention should be paid to the shift key, the caps lock key, and the ctrl key:

The shift key works like the corresponding key on normal typewriters. When the shift key is held down, pressing the A key yields an upper case A instead of a lower case a.

The caps lock key works almost like the shift lock key on typewriters. However, the caps lock key only locks the letters in upper case; you





still have to press the shift key to get the upper case variants of the other characters.

The ctrl (that is, 'control') key is another kind of shift key. Like the shift key it changes the meaning of any character pressed while the key is held down. However, keys pressed while the ctrl key is down will normally not cause anything to be displayed on the terminal screen. Normally only three keys will be used with the ctrl key, namely the C, S, and Q keys:

If the computer is writing a lot of text to the terminal, you may temporarily stop the the terminal by holding down the ctrl key and pressing the S key. When you want the terminal to continue, hold down the ctrl key and press the Q key.

A few programs may request you to enter a so-called 'end-of-file' in response to a question. An end-of-file is typed by holding down the ctrl key and pressing the C key.

Finally, note that the cursor need not be at the end of the line when you press the RETURN key.

### 3.3. The Edit Operation.

In many cases when you are requested to type something to the terminal, you will be typing onto a blank line on the screen. However, occasionally the line will not be blank when you are requested to type. Instead the computer will have placed some characters on the line for you. These characters will often be what the computer assumes you are most likely to type. In this case, what you have to do is change the contents of the line, if required, and press the RETURN key. The operation performed by the computer in this case is termed an Edit Operation.

Example: The computer displays the following line on the screen (the underscore represents the cursor):

```
Do you want to continue? YES
```

In this case, moving the cursor left and right may reveal to you that you are allowed to type different letters on top of the YES. If you



are content with the YES answer, merely press the RETURN key. If you want to answer NO to the question, type NO (followed by the space bar to remove the S from YES) and press the RETURN key.

## 4. Iounits.

Note: Not all users of the Unimax computer need know about iounits. It all depends on the kind of programs you will be running.

All input and output (commonly known as i/o) from the computer takes place between the computer and some device, such as a terminal, a printer, or a magnetic disk storage.

For all these devices Unirex uses the generic term 'iounits' (pronounced 'eye-oh-units'). This chapter will briefly describe the characteristics of each kind of iounit.

Each iounit connected to the computer has a name, the 'iounit name'. The first character of an iounit name is normally a colon. Generally, iounit names may be given as lower case or upper case letters at the user's discretion.

### 4.1. Terminals.

You are already acquainted with terminals. Terminals are used for direct communication with an operator. Terminals perform line-by-line or character-by-character i/o.

Each terminal has a number. The first terminal is number zero, the second terminal is number one, and so on. The iounit names of terminals consist of a colon, the letters 'term', and the number of the terminal given as one, two, or three digits.

The System Administrator may inform you about the numbering of the terminals at your installation.

Examples:    :term0  
              :term3  
              :term13  
              :term013

The two last iounit names designate the same terminal.



## 4.2. Printers.

A printer is normally, but not always, an output-only device. It is like a typewriter without a keyboard, and it is used to provide computer output on paper.

Each printer has a number. The first printer is number zero, the second printer is number one, and so on. The iounit names of printers consist of a colon, the letters 'print', and the number of the printer given as one, two, or three digits.

The System Administrator may inform you about the numbering of the printers at your installation.

Examples:    :print0  
              :print3  
              :print03  
              :print003

The three last iounit names designate the same printer.

## 4.3. Disks.

A disk is a magnetic device on which data can be stored and retrieved. Disks may be fast or slow devices, they may be removable or fixed. A special kind of disk is the so-called 'floppy disk' or 'diskette', which looks like a grammophone record.

Each disk has a number. The first disk is number zero, the second disk is number one, and so on. The iounit names of disks consist of a colon, the letters 'disk', and the number of the disk given as one, two, or three digits.

The System Administrator may inform you about the numbering of the disks at your installation.

Examples:    :disk0  
              :disk5  
              :disk12

The user is rarely allowed to perform i/o directly to or from a disk.

Normally the storage space on the disk is divided into several 'lumps' of data, called 'files'.

#### 4.4. Files.

Files are groups of data located on disks. A file may contain text or some other kind of data. Files are not necessarily permanently present on the disk: Files may be created or deleted as required, the size of a file may grow or diminish.

A disk may be partitioned into files in many different manners, called 'file systems'. Unirex is currently able to handle only one such file system, called 'Mikfile'. Mikfile was originally created for the SPC/1 micro computer from DDE. Disks with files created on an SPC/1 may be used directly on a Unimax using the Mikfile file system.

The Mikfile file sytem is described in greater detail in the Unirex System Operation Guide.

The iounit name for a Mikfile file consists of:

- The iounit name for the disk on which the file resides
- A slash (/)
- 1-8 characters called the file 'name'
- A hyphen (-)
- One character called the file 'type'

The disk iounit name may be omitted, in which case :disk0 is used.

The file type specifies the kind of data stored in the file. Many programs allow the user to omit the hyphen and the file type, in which case a default file type is assumed, depending on the program. Some programs actually forbid the presence of the hyphen and the file type.

Examples of valid Mikfile file names are:

```
:disk0/alpha-x  
/alpha-x  
/12345678-9  
:disk2/z-t  
:disk3/a.b-c
```

The first two iounit names designate the same file.

#### 4.5. Boxes.

Sometimes programs want to communicate with one another. For this purpose a running program may create a so-called 'box'. A box may be thought of as a fast file to which one program may write while another program is reading from it. Boxes are described in greater detail in the Unirex System Operation Guide.

The iounit names of boxes consist of a colon, the characters 'box', 'sysbox', or 'combox', a slash, and up to eight characters.

Examples of valid box names are:

```
:box/aragorn
:sysbox/strider
:combox/1a2b3c4d
```

#### 4.6. The null device.

One iounit in the system is called the null device. When data is written to the null device, the data is simply discarded. When a program tries to read from the null device, the device immediately signals end-of-file, that is, no more data available.

The iounit name of the null device is

```
:null
```

What, then, do you use the null device for? Well, many programs produce output which you from time to time feel is irrelevant. Suppose, for example, that you have a program that counts the number of lines in a file and at the same time writes the lines to an iounit. Assume that the program asks you to enter the name of the iounit to which it is to write the lines, and assume that for the present you are interested only in the number of lines in the file and don't want the listing of the lines. If you specify ':null' as the iounit to which the program writes the lines, the program will run faster and only the count will be produced; provided, of course, that the count is not output to the same iounit as the line list.

#### 4.7. Current Unit Prefix.

Let us assume that you have most of your files - or perhaps all of them - located on the disk `:disk3`. This means that all your file specifications start with the characters `:disk3/`; for example:

```
:disk3/alpha-k
:disk3/beta-l
:disk3/gamma-u
```

Writing `:disk3/` all the time can be very annoying. It would be much more easy if the computer knew, that your files were located on `:disk3`. Unirex has a feature known as 'current unit prefix', which meets this need. When a program starts executing on the computer, it is assigned a current unit prefix, which is a string of characters such as `:disk3/`. Now, when you specify an iounit name to your program, one of two things happen:

- 1) If the first character of the iounit name is a colon (:) or a slash (/), the iounit name is used as it is written.
- 2) Otherwise, the specified iounit name is appended to the current unit prefix before it is used.

If, for example, your program's current unit prefix is `:disk3/` and you specify an iounit name as `'eowyn-x'`, the actual iounit name used by the program is `:disk3/eowyn-x`. If, on the other hand, you specify an iounit name as `'/denethor-k'` or `:disk0/denethor-k'`, this iounit name will be used unmodified.

This means that if your program's current unit prefix is `:disk3/` your files

```
:disk3/alpha-k
:disk3/beta-l
:disk3/gamma-u
```

may be specified simply as:

```
alpha-k
beta-l
gamma-u
```

How do you specify the current unit prefix that you want? Well, that depends on what program you are running. When you log on to the terminal, your operator service program will be assigned an initial current



unit prefix. What this prefix will be, was specified by the System Administrator when he or she granted you access to the computer. Depending on the operator service program, you may or may not be able to change the current unit prefix. If your operator service program is capable of starting other programs, the new programs will inherit the current unit prefix of the program that started them.





## 5. Access Rights.

This chapter is about security. In many cases you may want to protect the contents of your files against other users. The reason for this may either be that you fear the some other user (or perhaps yourself) will accidentally or maliciously destroy the contents of your files. Or your files may contain confidential information which you do not want others to access.

To protect your files, and indeed all iounits, against unauthorized access, each iounit is assigned an 'owner number' and a set of 'protection bits'. The owner number is the user number of the person who owns the iounit; for files this will normally be the person who created the file. The protection bits control the access rights for various users of the computer. This will be described in greater detail below.

### 5.1. User Numbers.

Each user of the computer is assigned a user number by the System Administrator. A user number is a 4-digit hexadecimal number. By 'hexadecimal' we mean that the number may contain not only the digits 0-9 but also the letters a-f; the sixteen characters 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, and f are called 'hexadecimal digits'. Examples of valid user numbers are:

1234  
8a7f  
e004  
0488

The first two hexadecimal digits of each user number is called the group number of the user. In the above examples, the group numbers are 12, 8a, e0, and 04, respectively. Users with the same group number will typically be working on the same project or belong to the same department in a company.



## 5.2. Protection Bits.

The protection bits of an iounit specify who has access to that iounit. A 'bit' may be thought of as a switch that may be either off or on. Each iounit has 12 such 'switches'. The first three are discussed in the Unirex System Operation Guide.

The last nine protection bits are grouped as three 'triplets' of three bits each. The first set of triplets specifies the access rights of the owner of the iounit, the second set specifies the access rights of users with the same group number as the owner, the third set specifies the access rights of all others.

The three bits within each triplet have the following significance. If the first bit (the first 'switch') is on, the persons in question are allowed to read the contents of the iounit. If the second bit is on, the persons in question are allowed to write to the iounit, that is, change its contents. The third bit is relevant only for files containing programs. If this bit is on, the persons in question are allowed to execute the program contained in the file.

A typical set of access rights for a file may be: Read and write access to the owner of the file, read access to users with the same group number as the owner of the file, and no access rights for others.

How do you set and change the access rights of an iounit? When a file or box is created, the program creating the iounit assigns a set of protection bits to it. The owner of the iounit will normally be the user who executed the program. You may, however, change the access rights of the iounits that you own, provided that your operator service program allows you to do so. If your operator service program is the standard operator communications program, vox, you may execute the chaces program to perform this task. You are normally not allowed to change the owner number of a iounit.

What is said above about files and boxes applies also to terminals, printers, and disks accessed without a file system. The owner number and the protection bits assigned to these iounits are determined by the System Administrator.

### 5.3. Privileged Users.

One or more users of your computer, including the System Administrator, are 'privileged'. Privileged users are users within group number 00.

Privileged users have special rights within the computer. They may change the access rights of any iounit, including iounits they do not own, and they may change the owner of any iounit. Several other operations, such as aborting any program running on the computer or setting the internal clock of the computer, may be performed only by privileged users.



## 6. The Computer.

This chapter gives a very brief description of the internal structure of the Unimax computer. Although the normal user need not be aware of these things, a little understanding of the anatomy of the computer often helps in getting the most out of the machine.

Internally the Unimax computer really is up 16 computers integrated into one. More specifically, the Unimax computer contains up to 16 independent micro processors, each having their own memory. Some of these micro processors are dedicated to special tasks, such as writing to and reading from a terminal, while others are general purpose 'computers' that perform the actual computing for you. The latter are called MCUs, 'Main Computing Units'.

As described in chapter 3, when you log on to the Unimax the statement 'You are now logging on to processor number: 1' is presented, and you are allowed to change the number indicated. This number is actually the number of the MCU on which you are going to run your programs. The processor number suggested at log on time is the one which the computer thinks is currently least used.

### 6.1. System Crash.

Nobody is infallible. Error-free programs are as rare as oases in a desert, or perhaps even more so. Therefore it is inevitable that now and then an error in Unirex occurs. The system has been designed in such a way that it often is able to detect internal errors and stop, rather than continue erroneous execution. This is known as a system crash.

If you are working at terminal number zero (:term0) when a system crash occurs, you will see the following text on the terminal.

```
*****  
**** SYSTEM CRASH ****  
*****
```

```
INSERT FLOPPY DISK ON WHICH TO DUMP MEMORY  
NOTE: PREVIOUS DISK CONTENTS WILL BE DESTROYED!  
PRESS RETURN:
```



When you see this text, stop working immediately! Do not press the RETURN key. Contact the System Administrator, who will know what to do in this case.