49135/49136, 49161, 49414 Fall term 1999

Using the E data bar

Department of Information Technology, Technical University of Denmark
Michael R. Hansen, Jørgen Arendt Jensen, Morten Olsen, Hans Rischel and Peter Sestoft

These notes explain how to use the E data bar at DTU.

The data bar is in the basement of building 345 West: enter building 344 through the northern entry and follow the "`E data bar`" signs down to the basement. Some of the doors are locked, but just use your student registration card in the card readers to open the doors. The data bar consists of a small room with the printers `eps1` and `eps2` and a big room with PC's labeled `linux01`, `linux02`, ..., `linux38`. These PC use the operating system **Linux** which is *different* from the usual Windows system on PC's. Linux belongs to the UNIX family of operating systems which are widely used at universities.

Contents

1 Logging in

The acess to the computers in the DTU databar system requires an user id (usually the letter "`c`" followed by your student registration number) and a password. You will get your user id and password on a sheet of paper which is issued by the DTU administration.

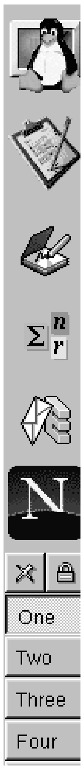
You start a session by logging in at a PC and you terminate the session by logging out. Logging in comprises the following steps:

1. Get the log-in box on the screen by moving the mouse. If this doesn't work then press the **Ctrl**, **Alt** and **BS** keys at the same time.
2. Move the cursor to the log-in box and type you user id and password each terminated by **return**. Note that the system does not echo your password in any way (even not by asterisks).
3. The desktop appears on the screen and log in is completed.

A session should always be terminated by logging out. You are NOT allowed to try to monopolize the PC by locking the screen. The PC must not be turned off as it is a multi-user machine where other users might be logged on.

2 Executing commands via the button-bar

The desktop environment consists of a *button-bar* to the right, some *desk top icons* to the left, and a *task bar* in the bottom of the screen. The button-bar gives an easy way for executing some commonly used commands. The picture at the right shows the buttons in the middle part of the button-bar. A button is actived by click'ing on the button (moving the cursor to the button and pressing the left mouse key):



Penguin creates an Xterm window. The window can be used to type any UNIX command.

Paper sheet starts the editor program Emacs in a new window. Emacs can e.g. be used for editing programs, and the SML compiler can be invoked from inside Emacs.

Note book starts the editor program Nedit in a new window. Nedit can e.g. be used for editing Matlab programs.

Math symbols starts Matlab in a new window.

Letter E starts Pine in a new window. Pine is used for sending and receiveing electronic mail.

N starts Netscape in a new window. Netscape is used to find and display information stored in the World Wide Web on the Internet.

Dagger performs log out. Pressing the left mouse key creates the Exit dialog box where you click on **Ok**.

One, Two, Three, Four are used to shift the physical screen between the four desktops.

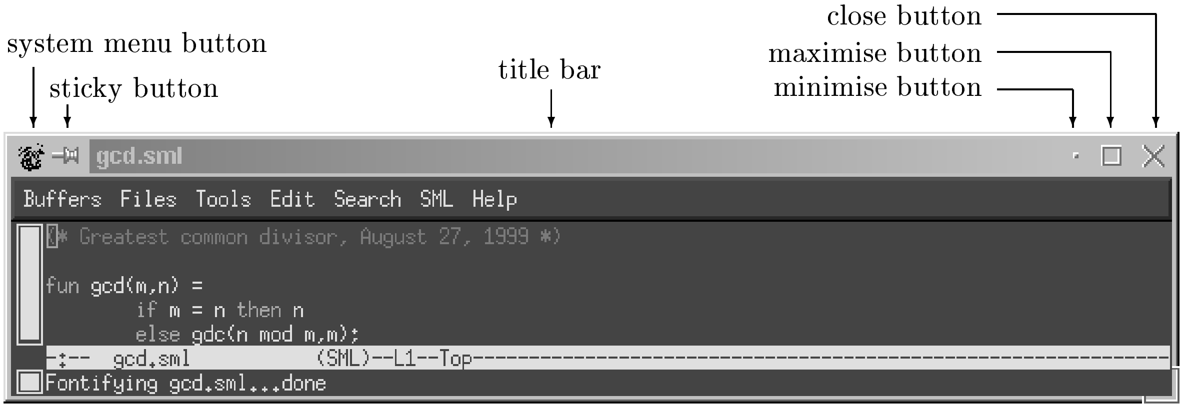


Figure 1: A window

3 Windows and icons

A window displays a separate dialogue with the operator. A window in the E databar has the general lay-out as shown in Figure 1 with a *system menu button*, a *sticky button*, a *title bar*, a *minimise button*, a *maximise button*, and a *close button* located at the top of the window.

A window is selected (and gets into the foreground) when the the left mouse key is click'ed in the window. A window can be moved around on the screen if the left mouse key is kept pressed down while the cursor is placed in the title bar. A window can be resized if the left mouse key is kept pressed down while the cursor is placed in one of the corners of the window.

Pressing the maximise button changed the window to extend over all of the screen, while pressing it once more gets the window back to the previous size. Pressing the minimize button shrinks the window to a small icon which is placed in the task bar. The task bar (as shown in Figure 2) is placed in the bottom of the desktop. A minimized window is restored by click'ing on the *icon* in the task bar.

Pressing the sticky button "sticks" the window to the desktop such that the window will be found in all four desktops (as selected by **One**, **Two**, **Three** and **Four**).

The close button is used to terminate the task in the window in a orderly fashion. A "dead" window can be killed by selecting kill in the system menu (move cursor to the system menu button in the upper left corner, press down the left mouse key, move to **kill** in the menu and release the key).

Figure 2: A task bar

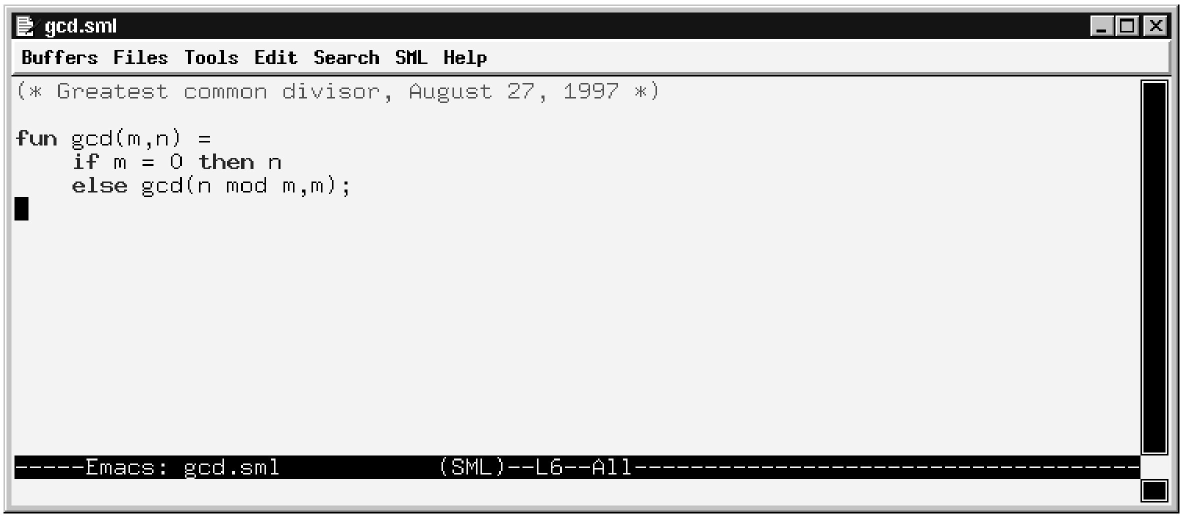


Figure 3: The Emacs editor with an SML program

4 Running SML programs (49135,49136,49161)

With Emacs, you can edit your SML program in one window and run an SML session in another window. Emacs provides simple commands for SML program layout, for loading the program into the SML session, and for repeating entries in the ML session.

Emacs is a powerful editor once you have learned to use it, but the commands may appear strange at first. The menus, however, make use of Emacs easier. An overview of Emacs commands is found in the *GNU Emacs Reference Card*. The composed keys `M-`x ("meta-x") and `C-`x ("control-x") for arbitrary key x are made as follows:

`M-`x	means:	hold down the "Alt" or "Alt function" key and press the key x.
`C-`x	means:	hold down the "Ctrl" key and press the key x.

This a brief step-by-step guide to editing programs.

1. Start Emacs by use of the emacs button in the button-bar. An empty Emacs window called `*scratch*` appears. Move the cursor to the Emacs window and click the left mouse button to select this window. Do not type your program in the `*scratch*` window.
2. To create an SML program buffer for a program called "`gcd`", type `C-x C-f`, then `gcd.sml`, then newline. (Alternatively, use the `Files` menu.)
   If the file `gcd.sml` exists on disk already, it is loaded into Emacs. Otherwise a new empty buffer is created. The suffix "`.sml`" is important: it puts the buffer in Sml mode.
3. Type and edit your program in the SML program buffer. Editing commands are described in the reference card.

Figure 4: The SML session window

4. Type `C-x C-s` to save the program in a file on the disk. (Alternatively, use the `Files` menu.)
5. To load the program into an SML session, type `C-c C-b` in the SML program buffer. Type new-line twice to answer the questions Ml command: mosml and Any args:.
   This creates an SML session, and loads the program into the SML session.
6. Type `C-c C-s` to see the output from the SML session. The SML session will be in a new window named mosml, as in Figure 4 on page 5.
7. Resize and move the windows (as described in Section 3) to separate the two windows on the screen.
8. If the program has syntax errors or type errors, the SML session prints messages about them. Use `` C-c ` `` to move the cursor to the position where an error has been located (on PC's the character `` ` `` is the upper case of the key  just to the left of the backspace key). Edit the program to correct the errors. When you have finished correcting the errors, go back to step 4 above. Typing `C-c C-b` again will load the corrected program into the existing SML session, and SML prints the types of the functions and values defined.
9. Select the SML session window by moving the cursor to the window and click the left mouse button. Enter some expressions to test your program. To go back to the SML program window, move the cursor to the window and click the left mouse button.
10. When you have finished editing and testing, terminate the SML session by typing `C-c C-d` in the SML session window.
11. To print out the program or the output from the SML session, select the appropriate window and activate Print buffer or Postscript print buffer in the Tools menu (move the cursor to Tools at the top of the window, press down the left mouse button, keep the button pressed down, move the cursor to the wanted item, and release the button). The printer name is `eps1`
12. To terminate Emacs, type `C-x C-c` or press the close button in the window (cf. Sec-

tion 3). If you have not saved your program file, it will ask “`Save file ...?`”, to which you should answer “`y`”.

Key kombinations like `C-c C-b` for sending a program to the SML compiler are specific for the *sml mode* of Emacs which is activated by the extension “`.sml`” of the file name. An overview of these special functions can be found in the `Sml` menu (click on `Sml` in the Emacs window).

If you get into a mess with Emacs, try `C-g` once or twice to interrupt it. If this does not help, type `C-x C-c` (or press the close button in the window) to terminate Emacs, and answer any questions it asks. If this does not help, then you could use “brute force” and select kill in the system menu of the window. This kills Emacs but the program file may get garbled.

5 SML system functions (49135, 49136, 49161)

This section describes a selection of built-in functions for controlling the Moscow ML compiler in interactive mode.

id	type	value	exception
`compile`	`string -> unit`	compile unit (*U*.`sml` or *U*.`sig`	`Fail`
`load`	`string -> unit`	Load unit and any units it needs	`Fail`
`loadOne`	`string -> unit`	Load unit only	`Fail`
`printVal`	`'a -> 'a`	Print value on `stdOut`	
`printDepth`	`int ref`	Limit printed data depth	
`printLength`	`int ref`	Limit printed list and vector length	
`quit`	`unit -> unit`	Quit the interactive SML system	
`system`	`string -> unit`	Execute operating system command	
`use`	`string -> unit`	Include file in program	
`verbose`	`bool ref`	Permit feedback from `compile`	

The entities `printDepth`, `printLength`, and `verbose` are set by an *assignment* e.g. the assignment:

```
printLength := 100;
```

ensures that the response from SML will print the first 100 elements of a list.

6 The editor Nedit (49414)

An alternative to Emacs is the editor Nedit, which also can be run from the button-bar. Nedit resembels a usual Windows editor with the same menus and short cut keys, and it can be used for making standard text files for Matlab and C programs.

7 Running matlab programs (49414)

Matlab is a powerful simulation language for manipulating vectors and matrices, and can be used for one-dimensional and two-dimensional signal processing. It is an interpretive language and commands are execute as they are entered at the keyboard.

The program is started by writing `matlab` in an xterm window or by pressing the Matlab button in the button-bar. A graphic window will appear shortly, if you are running under the X-windows system, and it indicates that graphs can be shown.

You can now enter commands and results will be presented after the calculation. A good introduction to the various facilities can be gained by writing `demo` and clicking on the different menus.

The small example below shows how to define a sinusoidal signal with 8 periods, plotting it and plotting the Fourier transform of the signal. Note the semicolon in the lines that suppresses the output of the calculate data in the window. Note also that the variables `t` and `y` are vectors, and that Matlab can handle complex numbers, as is returned by the function `fft`.

```
f0 = 4e6;
fs = 100e6;
t = (0:1/fs:8/f0);
y = sin (2*pi*t*f0);
plot (t,y)
title('Sinusoidal signal')
xlabel ('Time [s]')
plot((0:255)/256*fs, abs(fft(y,256)) )
title('Amplitude spectrum of signal')
xlabel ('Frequency [Hz]')
```

An explanation of the different commands can be obtained by writing `help` *command-name* like `help plot`. The contents of the graphical window can be printed on printer `eps1` using the `print` command.

Matlab can also be extended by writing m-files. These are textfiles that contains the commands as they are written at the keyboard. The commands shown above can be written into the file `sin_sig.m` using Emacs, and placed in the current directory. The commands are then executed by writing `sin_sig` at the Matlab prompt. Functions returning values can also be defined, see `help function`

Nedit is invoked by the command `edit`, when a new file should be created or `edit` *file-name*, when an existing file should be changed.

Matlab is terminated by the `exit` command.

8 File, directory and path

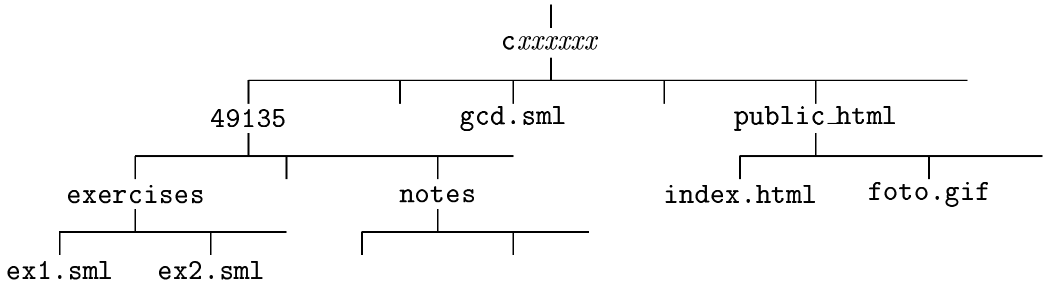


Figure 5: Possible file organisation for a user c*xxxxxx*

Files on disk are organised in directories. A directory may contain subdirectories and this gives a tree structure on the set of files. The private files for a user c*xxxxxx* may e.g. be structured as shown in Figure 5. This user has a *home directory* c*xxxxxx* containing subdirectories `49135` for the course 49135 and `public_html` for files related to the user's home page on the WEB. The home directory contains furthermore a SML program `gcd.sml`, while the directory `49135` contains a subdirectory `exercises` which contains SML programs `ex1.sml` and `ex2.sml`, etc.

References to a file is made relative to the current directory. Hence, if `49135` has been selected as the current directory, then the file `ex1.sml` in the `exercises` subdirectory is specified by the path:

```
exercises/ex1.sml
```

while the file `index.html` is specified by a path using the parent directory symbol "`..`":

```
../public_html/index.html
```

Other users may read e.g. the file `ex1.sml` by using a path containing the user name preceded by a tilde character (unless reading by other users has been prohibited):

```
~cxxxxxx/49135/exercises/ex1.sml
```

9 Some useful UNIX commands

The commands described in this section are executed by typing text lines in an `xterm` window (cf. Sections 2 and 3). The descriptions are brief, just giving the name of the command and a one-line explanation, but you have easy acess to more information by using the on-line documentation.

Online documentation

The online manual page system is a standard Unix feature, which you may use if you need information about a particular command. For example, to get a description of the Unix command `ls`, type "`man ls`" in an xterm window. Use the spacebar to move a page forward in the text, and use the arrow keys for scrolling.

Changing password

At the beginning you get a password which has been generated automatically. This must be changed to one you invent yourself. To change your password create an xterm window and use the `passwd` command:

1. Type `passwd` and newline to invoke the password changing command.
2. You will be asked to enter your old password, then to enter your new (future) password twice.

Using printers

`lpr`	print file on printer (e.g.: `lpr -Peps1` *file*)
`lpq`	display printer queue.
`lprm`	remove jobs from printer queue.

Creating and removing directories

`cd`	move to another directory.
`mkdir`	create new directory.
`rmdir`	remove directory.
`pwd`	show path to current directory

Using files

`cp`	copy file
`chmod`	change protection of file.
`find`	find files.
`less`	display contents of file.
`ls`	list attributes of file(s).
`more`	display contents of file.
`mv`	rename file.
`rm`	remove file.

Processes

`ps`	display current processes.
`kill`	kill process.
`free`	reports the amount of free memory.
`who`	display users logged in at the host.
`rwho`	display users logged in at the entire databar.

Using floppy-disks

You may copy files to/from a floppy disk in DOS format using the below commands. The floppy disk unit on the PC is called "`a:`".

`mcopy`	copy file to/from floppy disk.
`mdir`	display directory on floppy disk.
`mdel`	delete file on floppy disk.
`mformat`	format floppy disk.

Manipulation of text files

`diff`	compares two files and reports the differences.
`grep`	search file for pattern (regular expression).
`sort`	sorts textfiles.
`head`	prints first 10 lines of file.
`tail`	prints (surprise) last 10 lines of file.
`a2ps`	"pretty-prints" ASCII files and prints them on eps1.
`wc`	counts lines/words/letters in a file.

Redirection of input or output

To redirect the input or output for a program invoked by a command you use `<` and `>`. For example to redirect the output from a directory listing (command `ls`) to *outputfile*, you would do the following: `ls >` *outputfile*. To redirect the input from a file, you use `<`

Piplines

Piplines allow you to let the output of a program be used as input to another program. This is done in a command line containing several commands separated by bar characters "`|`". Each bar denotes a pipline which "pipes" the output of the command to the left of the bar to the input of the command to the right of the bar.

Example: let's assume that you have created a program that outputs textlines, and you want to se the first 15 lines containing the word "helloworld", you would execute the following: `myprogram | grep helloworld | head -n 15`

10 Desktop enviroments in the E-databar

The graphics subsystem in Linux is an implementation of *The X Window System* or just *X* for short. It implements the basic graphic and windowing primitives and it is totally network transparent (unlike the graphics system of another well-known operating system). This means that you can run an entire session, or just a single applications on any other computer in the DTU databar system from the PC in the E-databar.

A *windowmanager* works on top of the graphics subsystem and it implements the lay-out of windows (borders, buttons, etc.), how to move windows around, how to use menus, and so on. The term *desktop enviroment* covers X, a windowmanager and a suite of utilites, configuration programs, games, editors and so on.

KDE - The K Desktop Enviroment

This is one of the most mature and full featured desktops available for Linux. The developers of KDE seem to have used MS Windows quite a bit, and KDE is nauseatingly alike it. Of course if you use MS Windows at home, you should feel quite at home in KDE. Using so called "themes" you can change the look of windows, borders, backgrounds ect., giving opportunity for countless hours of fun doing "window shopping". This is the new default in the databar.

WindowMaker

WindowMaker is a windowmanager with a twist. It is designed to be like OpenStep/NextStep, the famous operating system and computer that unfortunatly failed back in the eighties. The "look and feel" is quite different from the other, but very effective.

GNOME - GNu Object Model Enviroment

GNOME was originally created because of some problems concering KDE. It is not quite as mature as KDE, and somewhat more unstable. I would recommend not using it yet, it is included in the databar for completeness. GNOME is much more configurable that KDE, and can be made to look like Star Trek on steroids. The "feel" can also be adjusted to your personal preference.

"Standard"

This is the desktop enviroment that have been used in the databar for the last few years. Its stable and fast, but very plain and extremly difficult to customize. It is included for older students that do not want to use a new one.

Other desktops

Almost all windowmanagers in existence are installed on the databar hosts. If you want to use some other than the 4 above, you are on your own! This means you will have to edit the file `.xsession` in your root directory to start the wanted windowmanager. If you really *want* another desktop and you know how to do it, this should pose no problem.

11 More about Linux

A lot of books exists on using Linux. All larger bookstores have them.

On the internet you can find a lot of Linux documentation. The E-databar homepage

`http://www.ebar.dtu.dk`

might contain these links in the near future.

The different desktop enviroments are found at the following home pages:

GNOME	`http://www.gnome.org/`
KDE	`http://www.kde.org/`
WindowMaker	`http://www.windowmaker.org/`

User-guides and HOWTO's can be found on `http://howto.linuxberg.com`

General computer and Linux related news on `http://slashdot.org`

Danish user groups

The Skåne-Sjællands Linux User Group (SSLUG) has a *lot* of material on their web-site. They also run mailing-lists/newsgroups where new-bies can get friendly help with all aspects of Linux. They also maintain a very nice user-guide, it is available on their website but can also be bought as an IDG booklet "Linux - Friheden til at vælge" (Available in Polyteknisk Boghandel)

SSLUG - `http://www.sslug.dk` "Linux - Friheden til at vælge" `http://www.sslug.dk/linuxbog`

Getting Linux on your own machine

The databar machines is running Redhat 6.0. If you want to get Linux on your home PC, there are several options. It is possible to buy it, eg. in "Polyteknisk Boghandel", where you will get a nice manual and other goodies. Sometimes the computermagazines includes a CDROM with Linux on, and the IDG booklets also include a CDROM, although some of them are Redhat 5.2. It is also possible (and *legal*) just to burn a copy of someones CD.

Its no problem if you want to use other distributions than Redhat, the only difference is that some files and scripts relating to the administration of the machine can be placed different. Other nice distributions include Debian (`http://www.debian.org` and Suse (`http://www.suse.dk`)

Linux can coexist with MS Windows on the same machine, and can read/write both FAT32 and NTFS.