DECoberon™ User's Guide

Release 4.2 (8 March 94)

Régis Crelier

Introduction

DECoberon™ is an implementation of Oberon™ for Digital Equipment Corporation DECstation. Both the programming language Oberon-2 and the Oberon System have been implemented. For a complete description of the Language and of the System, one should read the following books:

N. Wirth and M. Reiser: Programming in Oberon. Steps beyond Pascal and Modula-2.
Addison Wesley, 1992, ISBN 0-201-56543-9.
Tutorial for the Oberon programming language and concise language reference.

M. Reiser: The Oberon System. User Guide and Programmer's Manual.
Addison Wesley, 1991, ISBN 0-201-54422-9.
User manual for the programming environment and reference for the standard module library.

N. Wirth and J. Gutknecht: Project Oberon. The Design of an Operating System and Compiler.
Addison Wesley, 1992, ISBN 0-201-54428-8.
Program listings with explanations for the whole Oberon system, including the compiler for NS32000.

H. Mössenböck: Object-Oriented Programming in Oberon-2
Springer-Verlag, 1993, ISBN 0-387-56411-X.
Principles and applications of object-oriented programming with examples in Oberon-2.

The technical report 160 by H. Mössenböck, *Differences between Oberon and Oberon-2, The Programming Language Oberon-2*, is recommended too. It is present as on-line documentation:

Edit.Open Oberon2.Differences.Text
Edit.Open Oberon2.Report.Text

The following short tutorial describes the differences to the original implementation from a user's point of view.

DECoberon runs as a single Ultrix process on a color or monochrom DECstation. It has the form of an X Windows application using only one X window covering the whole screen. This window can be controlled by the Window Manager, i.e. it can be pushed, moved, resized or iconized, but the WM has no control on the internal layout of the window (tracks and viewers are not X windows). The dimensions of the user and system tracks are automatically defined when starting DECoberon and cannot be changed until the Oberon process terminates, even if the window is resized. Since DECoberon is a normal X application, it can be used remotely over a network.

Installation

DECoberon is distributed as a set of compressed tar files (see the accompanying readme file for a description of the contents). The uncompressed files have to be copied into a common directory (*/usr/local/Oberon* is recommended) and hence can be used by several users. This directory should me made readonly, in order to avoid accidental removal of files. DECoberon is started from a current directory using a symbolic link to the executable which name is *oberon*. The link can be defined as follows: *ln –s /usr/local/Oberon/oberon oberon*. The environment variable *OBERON* is used, when defined, as search path for files. Its default setting is *'. /usr/local/Oberon'*. Each user can define *OBERON* in his *.login* file (e.g. *setenv OBERON '. MyLib /usr/local/Oberon'*).

To improve the character drawing speed on the screen, the Oberon fonts should be installed as X fonts. This installation has to be done only once. An Oberon tool converts the given Oberon fonts into ascii files. These files describe the fonts in the portable BDF format. A font compiler (*bdftosnf*, *bdftopcf*, *dxfc*, ...), that should be present on each X station, then compiles these files and produces machine-dependent fonts (SNF or PCF format) that can be installed in a new directory on the server. To convert and install the fonts, see the text file *InstallXFonts.Tool* and follow the instructions in that tool.

Starting DECoberon

DECoberon is started from a terminal window using the executable *oberon*. The syntax is:
oberon [–h heapsizeinMB] [–x module command] [–b bootfile]
[–f fontmapfile] [–d displayname] [–g geometry] [–c]

where
- *heapsizeinMB* is the size of the allocated Oberon heap in MB (default: –h 4)
- the command *module.command* is executed (default: –x Oberon Loop)
- *bootfilename* is the name of the boot file (default: –b DECoberon.Boot)
- *fontmapfile* is the name of the font mapping file (default: –f "Fonts.Map")
- *displayname* is the name of the X server display used, e.g. –d name:0.0 (default: –d "")
- *geometry* gives the position and dimensions of the Oberon window, e.g. –g 800x600+100+100 (the default depends on the dimensions of the screen)
- the option *–c* allocates a private colormap of 256 colors (default: not set, 16 colors from 0 up to 15)

As in all X applications, the window needs to get the focus before being able to receive input from the keyboard. Just click the left mouse button with the pointer in the DECoberon window.

If DECoberon gets out of control (e.g. because of a looping command), press Ctrl-C in the terminal window (do not forget to set the focus first). Xlib is not reentrant, and hence should not be interrupted. To avoid this problem, the Ctrl-C is ineffective if pressed while the Oberon process is executing outside an Oberon procedure, else a trap viewer should appear and execution will resume in the central Oberon loop. If the system does not react on that interrupt after several tries, one can kill DECoberon by pressing Ctrl-\ in the terminal window.

System.Quit terminates the oberon process. It is the normal way to leave DECoberon.

Differences to the Original Oberon

The differences are due to the underlying hardware and software. Here follows a list of modules containing differences:

System

New commands:

– *Quit* to leave DECoberon
– *Execute command* to execute a (non-interactive) shell command
– *ChangeDirectory newdir* to change the working directory

Compiler

Supports the Oberon-2 language.
Implements fine-grained consistency checking of separately compiled modules, which allows module interfaces to be extended without requiring a recompilation of client modules (*Object Model*).
See *Compiler.Tool* for options.

X11

New module acting as interface to Xlib.

Display

Imports *X11*.
Procedures *DefCC*, *DefCP*, *DrawCX*, *FadeCX*, *InitCC*, *InitCP*, *SetCursor* are not implemented.
Procedure *SetMode* has no effect.
The display bitmap is not directly accessible (*Map* returns 0).
The cursor is controlled by the window manager and is still active when a command is executing.
Pattern format is different, use *Display.NewPattern* to define a pattern.
image[1] defines the bottom line of the pattern. image[0] is not used.

Display1

Module exporting additional drawing procedures.

Unix

New module acting as interface to Unix (system calls).
Unix.Result() may be called immediately after a non function system call.
If not zero, it gives the error number.
Unix.Done() may be called immediately after a system call, first call *Unix.Result()* if the error number is needed.
Do not use function system calls as parameters of calls.
Pointer to array in Oberon-2 is not compatible with pointer to array in C;
if compatibility is required, use pointer to record a: array end

Input

Input events are puffered.
Special keys:

F1	mark viewer
Esc, F2	escape
F3	break
F4	init colors
Select, Ctrl-J	line feed
Ctrl-L	redraw screen
Prior (Page Up or PrevScreen)	redraw for Draw (also ë)
Ctrl-Prior	home for Draw (also ç)

International characters can be typed using the usual compose sequence or an international keyboard. A third way of composing them is to type the letter and the diacritic in any order, and then press the function key F5 (or the key Do) in order to combine the two symbols into a diacritical character.
Valid diacritics are : { ’ ‘ " ^ , ~ }

Input.Time() returns the real time in milliseconds since DECoberon has been started.

Files
Exports *ChangeDirectory.*

Types
New interface module to the type system. Necessary to implement persistent objects.

Oberon
Time() returns *Input.Time()*

Printer
generates a postscript file named *Oberon.Printfile.ps* and sends it to the printer if the printer name is not *none*.

Kernel
GC may be called at any time, since local pointers (on the stack or in registers) are used as roots during the mark phase of the garbage collection.
Garbage collection is done on (open) array elements too.

Font Mapping File
Since small Oberon fonts may be difficult to read on high resolution displays, several font mapping files are provided: *Normal.Map, Medium.Map* and *Big.Map*. (See option –f above).
Normal.Map yields better printing results.

The option character is "\" instead of "/". This allows to use "/" in file names (Unix path names).

For further information or bug reports, please contact

The Secretary	
Institut für Computersysteme	Tel: (+41-1) 632 7311
ETH Zentrum	Fax: (+41-1) 261 5389
CH-8092 Zürich	e-mail: oberon@inf.ethz.ch
Switzerland	(crelier@inf.ethz.ch for bug reports)