

The SuperPascal Software Notes

PER BRINCH HANSEN ¹

*School of Computer and Information Science
Syracuse University, Syracuse, NY 13244, USA*

November 1993

Abstract: These notes describe the *SuperPascal* software, define the terms and conditions for its use, and explain how you compile the *SuperPascal* compiler and interpreter.

1 Definitions

1.1 Software

The *SuperPascal* software (hereafter *Software*) is educational software written by Per Brinch Hansen (hereafter *PBH*). The *Software* consists of the *Manuals*, *Programs*, and *Scripts* for the programming language *SuperPascal* invented by *PBH*. The *Software* is stored as text in 11 files (hereafter *Files*).

1.2 Manuals

The *Manuals*, written by *PBH*, are stored as L^AT_EX text in 3 *Files*:

- *report.tex*: “The programming language SuperPascal” [Brinch Hansen 1993a].
- *user.tex*: “The SuperPascal user manual” [Brinch Hansen 1993b].
- *notes.tex*: “The SuperPascal software notes” [The present notes].

1.3 Programs

The *Programs*, written by *PBH*, are a *SuperPascal* compiler and interpreter (hereafter *Compiler* and *Interpreter*). The *Programs* are based on the Pascal compiler and interpreter described and listed in [Brinch Hansen 1985]. The *Programs* are written in Pascal for Sun3 and Sun4 workstations running Unix.

The *Programs* are stored as Pascal text in 6 *Files* (hereafter *Program Files*):

- *common.p*: The common declarations used by the *Compiler* and *Interpreter*.
- *scan.p*: The *Compiler* procedure that performs lexical analysis.

¹Copyright ©1993 Per Brinch Hansen. All rights reserved.

- *parse.p*: The *Compiler* procedure that performs syntax, scope, and type analysis.
- *assemble.p*: The *Compiler* procedure that assembles interpreted code.
- *compile.p*: The *Compiler* program.
- *interpret.p*: The *Interpreter* program.

1.4 Scripts

The *Scripts* are Unix shell scripts stored as text in 2 *Files*:

- *sun3.user*: A shell script for compilation of the *Programs* on a Sun3 workstation under Unix.
- *sun4.user*: A shell script for compilation of the *Programs* on a Sun4 workstation under Unix.

2 Terms and Conditions

THE MANUALS ARE COPYRIGHTED BY PBH. THE PROGRAMS ARE IN THE PUBLIC DOMAIN. YOU CAN OBTAIN THE SOFTWARE BY ANONYMOUS FTP. THE SOFTWARE IS NOT GUARANTEED FOR A PARTICULAR PURPOSE. PBH SUPPLIES THE SOFTWARE "AS IS" WITHOUT ANY WARRANTIES OR REPRESENTATIONS AND DOES NOT ACCEPT ANY LIABILITIES WITH RESPECT TO THE SOFTWARE. YOU (THE USER) ARE RESPONSIBLE FOR SELECTING THE SOFTWARE, AND FOR THE USE AND RESULTS OBTAINED FROM THE SOFTWARE. YOUR USE OF THE SOFTWARE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS.

3 Software Limits

The *Program File common.p* (hereafter *Common Declarations*) defines common constants, types, functions, and procedures used by the *Programs*. The limits of software arrays are defined by common constants (hereafter *Software Limits*). If the *Software Limits* are too small for compilation or execution of a user program, these limits must be increased by editing the *Common Declarations* and recompiling the *Programs*.

4 Include Commands

The *Program File* *compile.p* contains the following *include* commands:

```
#include "common.p"
#include "scan.p"
#include "parse.p"
#include "assemble.p"
```

These commands ensure that Pascal compilation of the *Compiler* also includes the *Common Declarations* and the *Compiler* procedures.

The *Program File* *interpret.p* contains the *include* command:

```
#include "common.p"
```

This command ensures that Pascal compilation of the *Interpreter* also includes the *Common Declarations*.

5 Nonstandard Pascal

The *Programs* use the following nonstandard statements, which are Sun extensions of Pascal [Sun Microsystems 1986]:

Program File	Procedure	Nonstandard statement
compile.p	testoutput	rewrite(log, kind)
compile.p	codeoutput	rewrite(code, codename)
compile.p	firstpass	rewrite(errors, errorfile)
compile.p	firstpass	reset(source, sourcename)
interpret.p	readtime	t := clock
interpret.p	openoutput	rewrite(outfile, outname)
interpret.p	openinput	reset(inpfile, inpname)
interpret.p	start	reset(codefile, codename)

The rest of the *Program Files* conform to *ISO Level 1 Standard Pascal* [British Standards Institute 1982].

6 Program Compilation

When you have obtained the *Files*, the first step is to compile the *Programs*.

On a *Sun3* you compile the *Programs* by typing the Unix command:

```
csn sun3.user
```

The *Script* *sun3.user* contains the Unix commands:

```
echo Compiling Sun3 SuperPascal
pc -s -H -O -f68881 -o sc compile.p
pc -s -H -O -f68881 -o sr interpret.p
```

The *Programs* are compiled with the following Sun3 options:

- *-s*: Check the Pascal standard.
- *-H*: Check pointers (but not subranges).
- *-O*: Optimize the code.
- *-f68881*: Generate code for the Motorola 68881 floating-point processor.

On a *Sun4* you compile the *Programs* by typing the Unix command:

```
    csh sun4.user
```

The *Script sun4.user* contains the Unix commands:

```
echo Compiling Sun4 SuperPascal
pc -s -H -O -cg89 -o sc compile.p
pc -s -H -O -cg89 -o sr interpret.p
```

The *Programs* are compiled with the following Sun4 options:

- *-s*: Check the Pascal standard.
- *-H*: Check pointers (but not subranges).
- *-O*: Optimize the code.
- *-cg89*: Generate code for any Sun4.

The *-s* option causes the Sun Pascal compilers to display warning messages about the nonstandard Pascal statements used in the *Programs*.

A compilation of the *Programs* takes 3–5 minutes and produces two *Executable Files* [Brinch Hansen 1993b]:

- *sc*: An executable *Compiler*.
- *sr*: An executable *Interpreter*.

If you are not using *SuperPascal* on a Sun3 or Sun4, try the following if the *Programs* cannot be compiled directly:

- Change or omit the compilation options in the *Scripts*.
- Change or omit the nonstandard statements in the *Program Files*.
- Include the *Common Declarations* in each of the other *Program Files*. These *Program Files* can then be compiled separately and linked together.

References

- [1] Brinch Hansen, P. (1985) *Brinch Hansen on Pascal Compilers*. Prentice-Hall, Englewood Cliffs, NJ.
- [2] Brinch Hansen, P. (1993a) The programming language SuperPascal. School of Computer and Information Science, Syracuse University, Syracuse, NY.
- [3] Brinch Hansen, P. (1993b) The SuperPascal user manual. School of Computer and Information Science, Syracuse University, Syracuse, NY.
- [4] British Standards Institute (1982) *Specification for Computer Programming Language Pascal*. BS 6192.
- [5] Sun Microsystems (1986) *Pascal Programmer's Guide*. Mountain View, CA.