Analysis, Design and Implementation of an Interactive Pascal Compiler on Personal Computers

This project is submitted in partial fulfillment of the requirements for award of the

DIPLOMA IN COMPUTER STUDIES
of the
HIGHER TECHNICAL INSTITUTE

CS/111

Project Supervisor : Mr. Christos Makarounas, Bsc
External Assessor : Mr. Andreas Kattos

DESIGNED BY
Flangofas John Damianou Nikos

June 1994

Summary

Analysis, Design And Implementation Of An Interactive Pascal Compiler On Personal Computers

Damianou Nikos *Flangofas John*

This project was initiated as an effort towards studying the sophisticated subject of compiler construction in depth. As of the very start, it was considered a very difficult achievement to have this project completed within time limits. The four months of work-time provided seemed rather inadequate to allow "wishful thinking". This of course was also due to the fact that apart from the compiler itself, an editing environment with a full implementation of menus, editing capabilities, mouse support, on-line help and more, was also due. This part, however, was finally dealt with after the completion of the compiler, using a powerful object-oriented language especially designed to handle interactive applications, the Turbo Vision facility of Turbo Pascal.

What puzzled us the most, was the building of the compiler. We had to start from scratch since we had no knowledge whatsoever on the subject of compiler-construction. The first thing to do was to find the right books and begin study on compilers in general, focusing on specific techniques and principles that govern the construction of a compiler. The aspects concerned were far too many since they involved knowledge in the following aspects :

- A third generation programming language with extremely good understanding of pointers, memory and recursion manipulation.
- A very good understanding of the programming language for which the compiler was to be built; that is Standard Pascal, which is one of the most strongly typed languages including a very complicated syntax to built a compiler for.
- An extremely good knowledge of machine or assembly language.

Based on hard study we managed to derive a model for our compiler and outline the phases and techniques that should be followed in implementing the compiler. This resulted in two important benefits that were the keys for the completion of the project. The first one was the division of the compiler into clear-cut phases and passes - we followed the multi-pass concept instead of the single-pass - and the second one was the decision to use the code-interpretation technique with the building of a pseudo-machine,

for the completion of the last part of the compiler : code generation. This meant that our compiler would not produce executable code, but we could then avoid using assembly or machine language for its building. Our knowledge and time available wouldn't have made it possible to complete the project if we wanted to output assembly or machine code. Finally, we solved the problem of the programming language by using Turbo Pascal for the implementation of the project.

The building of the Standard Pascal Compiler was definitely an extremely difficult task, but it was surely worth the effort.

Contents