Programming systems for instruction

Author(s):
Schild, Rudolf

Publication Date:
1973

Permanent Link:

https://doi.org/10.3929/ethz-a-000084856 →

Dissertation No. 5140

PROGRAMMING SYSTEMS FOR INSTRUCTION

A Dissertation submitted

to the

SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of

Doctor of Mathematics

Presented by

RUDOLF SCHILD
Dipl. Phys. ETH
born May 6, 1941
from Grenchen (Kt. Solothurn)

Accepted on the recommendation of
Prof. Dr. N. Wirth
Prof. Dr. P. Läuchli

1973

7. C O N C L U S I O N

This study may be looked at from two different viewpoints:

7.1. T o o l s f o r E d u c a t i o n i n P r o g r a m m i n g

(a) The programming language must be suitable for the task. In order that the programmer's thoughts can be expressed naturally, the language must provide the possibility of structuring. This study shows that it is possible to use such a language in an interactive system. Even though the natural unit of information exchange in a time sharing environment is the line, it is possible to build a system in such a way that the compiler can be totally unaware of this unit, which from the point of view of the language is completely artificial.

(b) The beginning programmer must be given as much help as possible in his struggles with the new subject. He must be limited to a number of well understood concepts so that he is forced to organize his thoughts in a transparent and well-structured manner. All errors which can be detected by the system must be signalled to him; this goes especially for the error of using uninitialized variables. In case of an error, he must be given useful information about the status of his program. Care must be taken, however, not to overwhelm him with too much data from which he is then unable to extract the relevant information. The given information must be in the programmer's language, otherwise it is useless to him.

But even in cases where the system detects no error, the programmer should be given some information on the performance of his program (frequency counts), either to help him in finding a logical error causing incorrect output, or to assist him in improving a working program.

(c) Statistics should be kept to provide the teacher with information on the students' programming activities. However, individual programs will still have to be judged directly by the teacher.

7.2 Providing the Tools

Compiler systems must be extremely reliable. This is especially true for a batch as well as a multiprogram system, where faulty performance due to one user's program might hurt many others.

The user, particularly the beginner, must be able to have confidence in the compiler, knowing that any unexpected results his program produces are indeed due to the program and not caused by a compiler error.

Thus it is necessary to build a compiler system in such a way that errors are most unlikely to occur.

Both Pascal-S and Pascal-V were developed using the methods of structured programming and stepwise refinement [3],[18],[19]. This approach proves particularly valuable in the case of assembly language programming: The Pascal-V system was first refined down to the level of Pascal and tested in this form, and then still further refined until the assembly language level was reached.

In spite of the extensive testing which both systems are continually undergoing as they are used daily, so far only one error was found. This turned out to be due to a misinterpretation of the machine reference manual and was easily corrected.

Therefore we may conclude that the kind of programming we teach and the methods we use do lead to the production of reliable software and thus achieve our ultimate goal.