

การสร้างตัวแปลภาษาอววกเป็นภาษาซี



นายเฉลิมศักดิ์ ฉัตรดอกไม้ไพร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

พ.ศ. 2537

ISBN 974-583-964-7

ลิขสิทธิ์ของบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

I16884000

CONSTRUCTION OF AN AWK-TO-C TRANSLATOR

Mr. Chalernsak Chatdokmaiprai

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science

Department of Computer Engineering

Graduate School

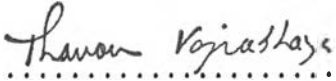
Chulalongkorn University

1994


ISBN 974-583-964-7


Thesis Title CONSTRUCTION OF AN AWK-TO-C TRANSLATOR
By Mr. Chalernsak Chatdokmaiprai
Department Computer Engineering
Thesis Advisor Associate Professor Somchai Thayarnyong


Accepted by the Graduate School, Chulalongkorn University in
Partial Fulfillment of the Requirements for the Master's Degree.



..... Dean of Graduate School
(Professor Dr. Thavon Vajrabhaya)

Thesis Committee


..... Chairman
(Dr. Yunyong Teng-Amnuay)


..... Thesis Advisor
(Associate Professor Somchai Thayarnyong)


..... Member
(Associate Professor Mandhana Prakansamut)


..... Member
(Mr. Charumatr Pinthong)

C317099 : MAJOR COMPUTER SCIENCE

KEY WORD: TRANSLATOR / PROGRAMMING LANGUAGE / AWK LANGUAGE / C LANGUAGE
CHALERMSAK CHATDOKMAIPRAI : CONSTRUCTION OF AN AWK-TO-C
TRANSLATOR. THESIS ADVISOR : ASSO. PROF. SOMCHAI THAYARNYONG.
69 pp. ISBN 974-583-964-7

This thesis presents the design and development of a software system that translates any AWK program into a functionally equivalent C program. The AWK-to-C translation system, which comprises an AWK-to-C translator and a library of subroutines that support the execution of the generated programs, has been constructed successfully on the Unix System V Release 4.0, using the Unix language development tools lex and yacc.


The purpose of the AWK-to-C translator is twofold. First, it serves as a faster alternative to the traditional AWK interpreters for processing AWK programs. Secondly, on the system with a C compiler but without an AWK language processor available, the AWK-to-C translator ported to the system, together with the C compiler, practically serves as a sole, complete implementation of the AWK language on that system.

A performance measurement suite was constructed to evaluate the performance of the translator-generated programs. The results indicate that in most cases, the generated programs run faster than their interpreted counterparts. The average speed improvement over the ten representative cases is 32%.

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....

สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์.....

ปีการศึกษา.....2536.....

ลายมือชื่อนิสิต..........

ลายมือชื่ออาจารย์ที่ปรึกษา..........

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม..........



พิมพ์ต้นฉบับบทคัดย่อวิทยานิพนธ์ภายในกรอบสี่เหลี่ยมนี้เพียงแผ่นเดียว

เฉลิมศักดิ์ ฉัตรดอกไม้ไพร : การสร้างตัวแปลภาษาอว๊กเป็นภาษาซี (CONSTRUCTION OF AN AWK-TO-C TRANSLATOR) อ. ที่ปรึกษา : รศ. สมชาย ทยานขง, 69 หน้า. ISBN 974-583-964-7

วิทยานิพนธ์นี้มีจุดประสงค์เพื่อออกแบบและพัฒนาระบบซอฟต์แวร์เพื่อใช้ในการแปลโปรแกรมภาษาอว๊กเป็น โปรแกรมภาษาซีที่ทำงานสมมูลกัน ระบบซอฟต์แวร์สำหรับการแปลดังกล่าวนี้ประกอบด้วยโปรแกรมตัวแปลภาษาอว๊กเป็นภาษาซี และคลังซบรูทีนที่จะสนับสนุนการทำงานของโปรแกรมที่ผลิตจากตัวแปลภาษา ระบบซอฟต์แวร์ทั้งหมดได้ถูกพัฒนาและทดสอบบนระบบปฏิบัติการ Unix System V Release 4.0 โดยใช้โปรแกรมที่เป็นเครื่องช่วยในการผลิตตัวแปลภาษาคือ เล็กซ์ (lex) และ แย็กค์ (yacc)

เป้าหมายการใช้ตัวแปลภาษาอว๊กเป็นภาษาซีมีสองประการ ประการแรก ระบบการแปลภาษาอว๊กเป็นภาษาซีนี้ เป็นอีกทางเลือกหนึ่งในการประมวลผลภาษาอว๊ก ซึ่งจะทำงานเร็วกว่าการใช้ตัวแปลคำสั่งภาษาอว๊ก (AWK interpreter) ที่ใช้กันมาแต่เดิม ประการที่สอง ในระบบคอมพิวเตอร์ที่ปราศจากตัวประมวลผลภาษาอว๊ก แต่มีตัวแปลโปรแกรมภาษาซี (C compiler) อยู่ นั้น เราสามารถจะใช้ระบบการแปลภาษาอว๊กเป็นภาษาซีร่วมกับตัวแปลภาษาซีที่มีอยู่ทำการประมวลผลโปรแกรมภาษาอว๊กในระบบคอมพิวเตอร์นั้นได้ทันที โดยไม่ต้องสร้างตัวประมวลผลภาษาอว๊กโดยตรงแต่อย่างใด

เนื่องจากมีเป้าหมายในการปรับปรุงความเร็วในการประมวลผลภาษาอว๊กดังกล่าว ชุดทดสอบประสิทธิภาพในด้านความเร็วของโปรแกรมที่ผลิตจากตัวแปลภาษาได้ถูกสร้างขึ้นเพื่อประมวลผลการทำงานของระบบการแปลภาษาอว๊กเป็นภาษาซี ผลที่ได้พบว่า โดยเฉลี่ยแล้ว โปรแกรมที่ผลิตจากตัวแปลภาษาอว๊กเป็นภาษาซีจะทำงานเร็วกว่าการใช้ตัวแปลคำสั่งภาษาอว๊กประมาณ 32 %

ภาควิชา วิศวกรรมคอมพิวเตอร์.....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์.....
ปีการศึกษา 2536.....

ลายมือชื่อนิสิต
ลายมือชื่ออาจารย์ที่ปรึกษา
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม



ACKNOWLEDGEMENTS

First and foremost, I wish to express my gratitude to Associate Professor Somchai Thayarnyong, my thesis advisor, for his guidance, suggestions, and time that helped make this thesis possible. I am also grateful to Ajarn Dr. Yunyong Teng-Amnuay who has been teaching the course 171312 Programming Language Compilers in which I enrolled in 1990 and from which I gained much knowledge about language translation needed to do this project.

During the tough time when I had to devote most of my energy to the project, Associate Professor Dr. Pateep Methakunawuthi gave me much needed financial aid by hiring me to do very flexible part-time job for which I feel deeply indebted to her.

Mr. Thanee Chavasirikultol and Mr. San Skulratanakulchai helped make it possible for me to set up a Unix system to do the entire project development. Mr. Chaiyasut Jongtavonwitaya and Miss Gesorn Soonsin let me borrow their software for writing the thesis documentation. Thank you Sear, Pee San, Lek, and Yai.

Also, I would like to specially thank Miss Kannikar Boonsumrej, my girlfriend, for her understanding and her believing in me no matter what. Without her, it must have been much harder for me to sail through this unusually intensive period of software development and still remain sane.

And, finally, I wish to thank my parents for their upbringing and their unrelenting support for my education. It's a great pity that they could not live long enough to see my graduation. Nevertheless, I dedicate this thesis to them.



CONTENTS

	Page
Abstract (English).....	iv
Abstract (Thai).....	v
Acknowledgements.....	vi
List of Tables.....	ix
List of Figures.....	x
Chapter	
I. Introduction.....	1
- Background and Rationale.....	1
- Objectives.....	2
- Scope and Limitations.....	3
- Development Procedure.....	3
- Expected Benefits and Applications.....	4
- Literature Review.....	5
- AWK Language.....	5
- C Language.....	5
- Programming Language Translation.....	5
- Unix Language Development Tools.....	5
II. Overview.....	7
- Components of the AWK-to-C Translation System.....	7
- Making an Executable Program from an AWK Program.....	7
- Overview of the System Development.....	8
- Development Environment.....	8
- Software Development Tools Used.....	9
III. Design of the Run-time Organization.....	11
- Storage Classes for AWK Expressions.....	11
- Implementation of AWK Variables.....	12
- Data Structure.....	12
- Classification of AWK Variables.....	14
- Implementing Global Scalar Variables.....	15
- Implementing Global Array Variables.....	18
- Implementing Local Variables.....	22
- Implementation of AWK User-defined Functions.....	22
- Function Class and The FnType Structure.....	22

- Function Invocation.....	22
- Parameter Passing Mechanism.....	23
- Run-time Tables.....	23
- Table of Global Variables.....	24
- Table of Input/Output Files and Pipes.....	24
IV. Development of the Translator.....	26
- Components of the Translator.....	26
- The Symbol Tables.....	27
- The Keyword Table.....	28
- The Global-name Table.....	28
- The Parameter-name Table.....	29
- The Main Routine.....	30
- The Scanner.....	30
- Development of the Scanner.....	30
- Token Definitions.....	32
- The Parser/Code-Generator.....	35
- Development of the Parser/Code-generator.....	35
- The Grammar for AWK Used by the Parser.....	35
- Interface Between the Scanner and the Parser.....	36
V. System Testing and Performance Measurement.....	38
- System Testing.....	38
- Performance Measurement.....	39
VI. Conclusions.....	41
- Summary of Results.....	41
- Suggestions for Further Development.....	42
- Performance of the Generated Program.....	42
- Portability.....	43
References.....	44
Appendix A. The AWC-to-C Translator User Manual.....	45
Appendix B. A Yacc Grammar for AWK Used by the Parser.....	48
Appendix C. A Translation Example.....	56
Biography.....	69

LIST OF TABLES

Table	Page
3.1 Storage classes for AWK expressions.....	11
3.2 Fields in the VarType structure.....	13
3.3 Variable type.....	13
3.4 Initial values of the VarType nodes for built-in variables....	17
3.5 Fields in the ArrElem structure.....	19
3.6 Fields in the VarInfoNode structure.....	24
3.7 Fields in the FileInfoNode structure.....	25
4.1 Translation-time categories of AWK identifiers.....	28
4.2 Fields in the Kword structure for keyword-table entries.....	28
4.3 Fields in the NameTabNode structure for the global-name table entries.....	29
4.4 Tokens recognized by the scanner.....	33
5.1 Execution times of test programs.....	39

LIST OF FIGURES

Figure	Page
2.1 Making an executable program from an AWK source program.....	8
3.1 Example of an ArrElem node.....	19
3.2 Data structure for AWK associative arrays.....	21
4.1 Components of the AWK-to-C translator and their relationships.	27
4.2 Algorithm for the translator's main routine.....	30
4.3 The scanner development process.....	31
4.4 Algorithm of the scanner routine.....	32
4.5 The parser development process.....	35
4.6 How the scanner and the parser routines work together.....	36
5.1 Illustration of the test procedure.....	38