CS420 컴파일러 디자인

교수: 이 광근 (compiler/~kwang)

강의: 화목 9:00am-10:30pm, 전산동 1101호

1999 봄

1 목표

Compilation process is an essential component in almost all programming systems. Every software that has a high-level language as its interface requires some form of compilation. Examples are compilers for programming languages, data base query languages, type-setting programs, logic synthesis systems, and etc.

This course has two goals: to provide students with principles of compiler design and to help students acquire implementation techniques. These two goals will be evenly emphasized.

2 무대장치

I recommend ML as the implementation language of your project compiler as well as of your homeworks. ML programming environments (SML/NJ and OCaml) for Unix(`cain`) is provided.

- TA: 정 재윤 `jay@compiler`, 주 상현 `hart@compiler`, 양 석호 (가나다 순)
- Web: `compiler/~kwang/420/99`
- BBS: `noah:course/cs420`

3 프로젝트

40% of your grade is determined by the performance of your project compiler.

- Project compiler must be able to translate a source program in S into an equivalent program of target language T.
- The performance of the compiled code is measured by our T interpreter, which records memory consumption and the number of instructions executed.
- We will provide you with a T debugger by which you can debug your compiler by investigating the execution of the generated T code.
- For the project, you must work in a group. Members in a group is 1 or 2. Every member of a group will receive equal points for the project.

4 숙제

Remaining 60% of your grade is determined by your homeworks.

- Homeworks are given on Thursday in class and due to next Thursday 24:00, to be submitted via computer.
- Every exercise in homeworks worths 10 points each. If your submission is late in 24 hours, then 10% off your points. If late in 24-48 hours, then 20% off. If late in 168 hours (1 week), then 50% off. We can't accept submissions later than 1 week.
- In case of homeworks, copying code is acceptable. When you use an existing code, however, you must report its author *and* demonstrate your improvement. Those who are cited by other people will get extra points.
- If you failed to show this honorary behavior, you are considered stealing codes from your friends and receives -10 points times the number of exercises in the particular homework.

5 내용및 진도

3/02	(1 wk)	overview (issues in compiler design)
3/09	(2 wks)	introduction to ML (implementation language)
3/23	(2 wks)	syntactic analysis (lexer and parser)
4/06	(2 wks)	target and source languages T & S (synatx and semantics)
4/20	(1.5 wks)	type checking (semantic verification)
4/29	(2 wks)	translation (frame, stack, and heap)
5/13	(2 wks)	optimization (register allocation, static analysis)
5/27	(1.5 wk)	module system (separate compilation, linking)
6/08	(1 wks)	run-time system (garbage collection)
6/24	(1pm)	project contest

6 책

교재 *Modern Compiler Implementation in ML*, 2nd Edition, Andrew Appel, Cambridge University Press, 1998

참고 *Compilers: Principles, Techniques, and Tools*, Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Addison-Wesley Publishing Co., 1986

참고 *Little MLer*, Mattias Felleisen and Daniel P. Friedman, MIT Press, 1998

Other useful materials will be available on-line in the course web page.