

Imperative language IFJ17 compiler implementation

Semestral project for subjects IFJ and IAL

Team 006, variant II

BRNO UNIVERSITY OF TECHNOLOGY

December 11, 2017

Team 006

Daniš Tomáš

xdanis05

Goldschmidt Patrik

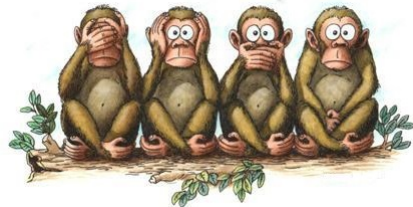
xgold00

Hamran Peter

xhamra00

Pšenák Kamil

xpsena02



Project structure

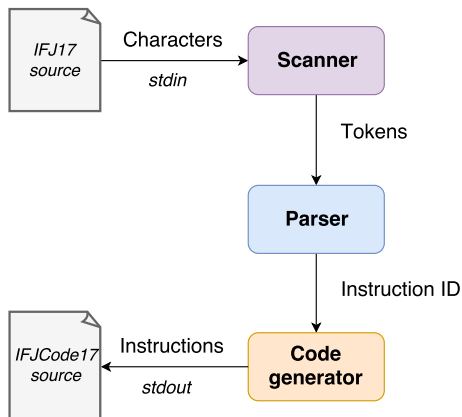


Figure: Compiler diagram

Lexical analyzer

- Input file → tokens
- Per-call basis
- Finite state automaton
- Error detection on unrecognized token
- Error recovery
 - Returns invalid tokens
 - Line counting

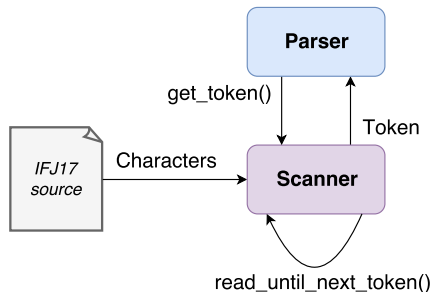


Figure: Scanner functionality

Syntactic and semantic analyzer

- Validates syntax and semantics
- Syntax-directed translation
- Recursive descent combined with PSA
- Syntactic and semantics analysis are run concurrently
 - Semantics stack
 - Symbol table
 - Global control variables

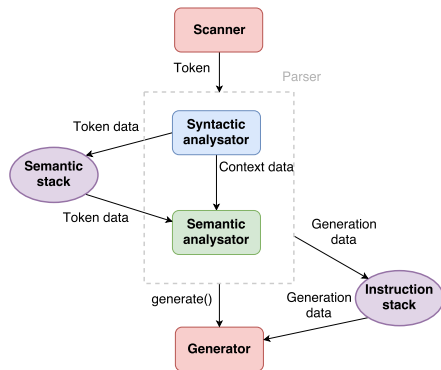


Figure: Parser functionality

Code generator

- Generates 3-address code
- Per-call basis
- Unique names → internal counter
- Generation process:
 - 1 Parser sets data and pushes it to the stack
 - 2 Parser calls generator routine with a specific instruction ID
 - 3 Generator adds instruction to the list
 - 4 Instructions are printed to stdout

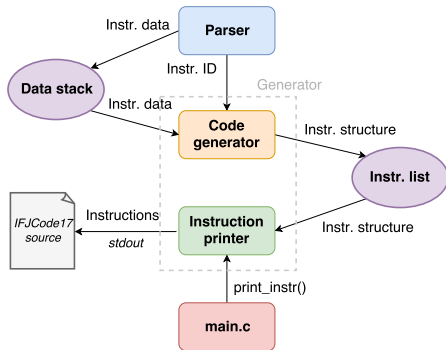


Figure: Generator functionality

Algorithms and data structures

- Symbol table
 - Hash table
 - FNV-1a 32-bit
 - Size: 300 elements
- PSA stack
- Instruction list
- Dynamic string
- Safe malloc
- Variable depth stack

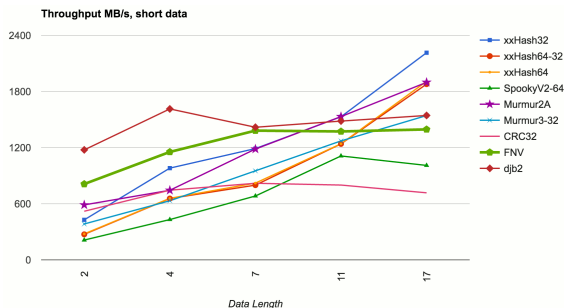
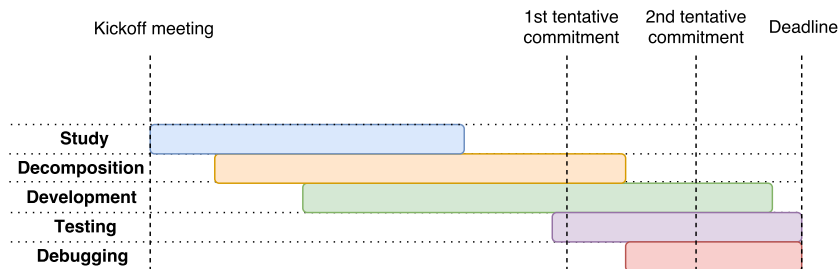


Figure: Hash functions performance [1]

Team organization

- Meetings on weekly basis
- Coding style adapted from CESNET's Nemea project
- Version control: `git`
- Agile software development techniques



Development problems

Things looked bad at first ...

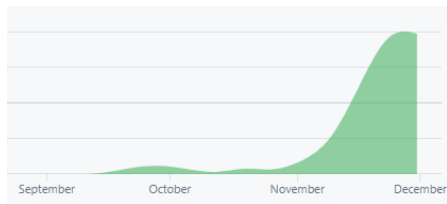


Figure: Sudden burst of inspiration

Bugfixing is not always easy ...



POINTERS DOOOOOOMED US ALLLLLL!!!!

hamyhamz committed 5 days ago

When you think you got it ...



SIGSEGV not fixed

hamyhamz committed 5 days ago

17:25



SIGSEGV in loops fixed

hamyhamz committed 5 days ago

17:18

Conclusion

- Great experience
- 13157 lines in 25 files
- Functional compiler for subset of FREEBasic language
- Benefits
 - Optimizations
 - Standard C99 → multiplatform
 - Interface documentation
- Downfalls
 - Recursive approach - long files may fill the buffer
 - Memory usage

Q&A





PRANCKEVIČIUS Ian. Hash functions all the way down.
[online], August 2016. Available at:
[https://aras-p.info/blog/2016/08/02/
Hash-Functions-all-the-way-down/](https://aras-p.info/blog/2016/08/02/Hash-Functions-all-the-way-down/). Seen
10.12.2017.