Iteratees in C

the lightning talk

pesco @khjk.org

30C3, Hamburg, 27-30.12.2013

Wat?

- Iteratees are stream processors.
- Programming model / API
  - to allow reasoning about I/O
- Origin: functional programming
- Challenge: Do it without first-class functions!
  - cf. Hammer

Security

- High level
- “Declarative”
- Be formal about accepted input
- Modular: reduce unmapped interactions

⇒ Avoid weird machines

- Cf. langsec

Case Study: Word Count

```
Iteratee word_ = bind_(dropws, dropword);
Iteratee countwords = wrap(decode(word_), count);

Iteratee it = apply(enumf(stdin), countwords);
uintptr_t nwords = (uintptr_t)finish(it);
```

Case Study: Word Count

```
Iteratee word_ = bind_(dropws, dropword);
Iteratee countwords = wrap(decode(word_), count);

Iteratee it = apply(enumf(stdin), countwords);
uintptr_t nwords = (uintptr_t)finish(it);
```

Case Study: Word Count

```
Iteratee word_ = bind_(dropws, dropword);
Iteratee countwords = wrap(decode(word_), count);

Iteratee it = apply(enumf(stdin), countwords);
uintptr_t nwords = (uintptr_t)finish(it);
```

Benchmark

- "rockyou" password list
  - 14.344.392 lines
  - ~14.44M words
- `wc -w`
  - 3.8s real 3.6s user 0.1s sys
  - ignores non-ASCII
- `./iter` (`main` = `test4`)
  - ~~9.2s real 8.5s user 0.7s sys~~
  - 3.7s real 3.6s user 0.1s sys
  - total allocation: 600MB (over whole runtime)
  - peak memory use: 3MB (concurrent)

PoC Implementation

- Basic iteratees
- Input from file descriptor
- "`decode`" combinator
- UTF-8 decoder
- Several simple test examples
  - word count, line count, UTF-8 character count, . . .
- Automatic memory management
  - uses standard `malloc`/`free` for arenas
  - x86 (32-bit) only right now (needs to know registers)
- ~1500 lines alltogether

Future Work

- More memory management options
- A larger case study
- Flesh out a proper library/API
- Recursive-descent parser combinators
- Iteratee API for Hammer
- . . .

Pointers

- PoC repo: `http://code.khjk.org/citer/`
  - code
  - slides
  - slides (30min talk)
- Feedback: `pesco@khjk.org`