

# jist

an operating  
system for MIPS

[www.github.com/timtadh/jist](http://www.github.com/timtadh/jist)

Dan DeCovnick, Tim Henderson, Steve Johnson

# Major Components of Jist

- \* Memory manager
- \* Stack manager
- \* Interrupt handler
- \* Standard library (I/O interface)
- \* Preprocessor
- \* Hunt the Wumpus

# Memory Manager

- \* Handles a heap for each program
  - \* Fully functional compacting heap
  - \* FULLY FUNCTIONAL COMPACTING HEAP!
    - \* init, alloc, free, compact, get, put
- \* Does not claim its own memory
  - \* Memory claimed by calling program using sbrk
- \* We would use virtual memory...
  - \* ...but we couldn't figure out how to get spim to use the TLB.

# Stack Manager

- \* Keeps track of each program's context
  - \* Registers, stack pointer, frame pointer
- \* Might implement memory protection...

# Interrupt Handler

- \* Replaces spim's included exception handler
- \* Lives in kernel space
- \* Pre-emptive round robin scheduler

# Standard Library

- \* Kernel mode requires memory-mapped IO
  - \* i.e. no IO syscalls allowed
  - \* Had to implement all IO using spim's memory-mapped IO interface
- \* stdlib.s contains functions to handle IO
  - \* read\_char, print\_char, readln, println
  - \* read\_int
  - \* printf

# Preprocessor (MPP)

- \* `#include stdlib.s`
- \* `#define set_to_zero [global]`
  - \* `add %1 $zero $zero`
  - \* `"set_to_zero $a0"`
  - \* Macros are mostly recursive
- \* Register aliasing
  - \* `@my_alias = $t0`
  - \* Self-documenting assembly code!
- \* Scoping
  - \* Makes labels and aliases local to surrounding scope
- \* All jist programs are statically compiled in, so MPP performs introspection and generates code



# Hunt the Wumpus

- \* Easily the most complex component of jist
- \* Complex cave system of 20 rooms
- \* Two scary pits to fall into
- \* Two scary bats to carry you to a random room
- \* ONE HUNGRY SLEEPY WUMPUS!