PICO RUBY

Episode II
ATTACK OF THE RAKE

Ruby Association Grant 2021 Report

hasumikin>&Monstarlab

July 12, 2022

Arts-Wallpapers.com

Episode Ⅱ

ATTACK OF THE RAKE

It is a patient time to the PicoRuby. Although the PRK Firmware has been released and it brought the First Prize of the Fukuoka Ruby Award and a Final nomination to the Ruby Prize, problems are piling up such as a lack of syntax rules and an inconsistent build system with mruby.

Hasumikin, an apprentice to a Rubyist, is strugglingly writing the compiler to compile mrblib of mruby under the coaching of Master MATZ to restore freedom to the micon galaxy....

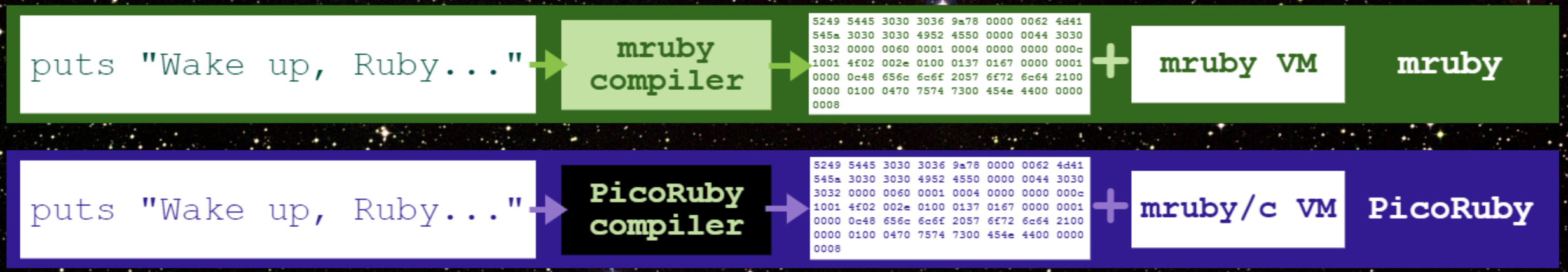
Arts-Wallpapers.com

What happened in Episode I



Arts-Wallpapers.com

What happened in Episode I



PicoRuby compiler for one-chip microcontrollers

Arts-Wallpapers.com

What happened between Episode I & II



What happened between Episode I & II



PRK Firmware

Keyboard is Essentially Ruby

Hitoshi HASUMI

RubyKaigi Takeout 2021 | Sep 10th

Arts-Wallpapers.com

What happened between Episode I & II

e.g.) picoruby/prk_crkbd



What happened between Episode I & II



What happened between Episode I & II

Unlike QMK Firmware,
You don't need
to **compile**
anything!

Arts-Wallpapers.com

What happened between Episode I & II

```
kbd = Keyboard.new
kbd.init_pins(
  [ 29, 28, 27, 26, 22, 20, 23 ],
  [ 3, 4, 5, 6, 7, 8, 9]
)

kbd.add_layer :default, %i[
  KC_ESCAPE KC_Q    KC_W    KC_E        KC_R    KC_T                                KC_Y    KC_U      KC_I     KC_O     KC_P      KC_MINUS
  KC_TAB    KC_A    KC_S    KC_Đ        KC_F    KC_G                                KC_H    KC_J      KC_K     KC_L     KC_SCOLON KC_BSPACE
  KC_LSFT   KC_Ž    KC_X    KC_C        KC_V    KC_B      KC_NO     KC_NO           KC_N    KC_M      KC_COMMA KC_ĐOT   KC_SLASH  KC_RSFT
                            KC_NO       KC_LALT KC_LCTL   LOWER_SPC RAISE_ENT SPC_CTL KC_RGUI   KC_NO
]
kbd.add_layer :raise, %i[
  KC_GRAVE  KC_EXLM KC_AT   KC_HASH     KC_ĐLR  KC_PERC                             KC_CIRC KC_AMPR   KC_ASTER KC_LPRN  KC_RPRN   KC_EQUAL
  KC_TAB    KC_LABK KC_LCBR KC_LBRACKET KC_LPRN KC_QUOTE                            KC_LEFT KC_ĐOWN   KC_UP    KC_RIGHT KC_UNĐS   KC_PIPE
  KC_LSFT   KC_RABK KC_RCBR KC_RBRACKET KC_RPRN KC_ĐQUO   KC_NO     KC_NO           KC_TILĐ KC_BSLASH KC_COMMA KC_ĐOT   KC_SLASH  KC_RSFT
                            KC_NO       KC_LALT KC_LCTL   AĐJUST    RAISE_ENT SPC_CTL KC_RGUI   BOOTSEL
]

kbd.define_composite_key :SPC_CTL, %i[KC_SPACE KC_RCTL]

kbd.define_mode_key :RAISE_ENT, [ :KC_ENTER, :raise, 150, 150 ]

rgb = RGB.new(0, 9, 0, false)
rgb.effect = :breath
kbd.append rgb

kbd.output_report_changed do ¦output¦
  rgb.hue = (output & Keyboard::LEĐ_CAPSLOCK > 0) ? 100 : 0
end

kbd.start!
```

Arts-Wallpapers.com

What happened between Episode I & II

- PRK Firmware
  - Compiling and applying a `keymap.rb` on a microcontroller without rebooting

Arts-Wallpapers.com

What happened between Episode I & II

- PRK Firmware
  - Compiling and applying a `keymap.rb` on a microcontroller without rebooting
- IRB in keyboard
  - Any text editor becomes an IRB

Arts-Wallpapers.com

What happened between Episode I & II

- PRK Firmware
  - Compiling and applying a `keymap.rb` on a microcontroller without rebooting
- IRB in keyboard
  - Any text editor becomes an IRB
- Presentations available on YouTube
  - Japanese -> RubyKaigi Takeout 2021
  - English -> RubyConf 2021

Arts-Wallpapers.com

Targets in Episode II



Targets in Episode II

- Rake
  - Integrating PicoRuby with mruby's build system

Arts-Wallpapers.com

Targets in Episode II

- Rake
  - Integrating PicoRuby with mruby's build system
- Syntax
  - Improving parser and generator to compile mrblib

Arts-Wallpapers.com

Targets in Episode II

- Rake
  - Integrating PicoRuby with mruby's build system
- Syntax
  - Improving parser and generator to compile mrblib
- Presym
  - Embedding mrblib in ROM

Arts-Wallpapers.com

Targets in Episode II

- Rake
  - Integrating PicoRuby with mruby's build system
- Syntax
  - Improving parser and generator to compile mrblib
- Presym
  - Embedding mrblib in ROM
- MicroRuby
  - PicoRuby compiler + mruby VM

Arts-Wallpapers.com


RAKE

Rake

- mruby takes advandage of Rake to be built
  - Fairly complicated (for me)
  - Too hard to migrate to Make (at least I can't)

Arts-Wallpapers.com

Rake

- mruby takes advandage of Rake to be built
  - Fairly complicated (for me)
  - Too hard to migrate to Make (at least I can't)
- PicoRuby used to use Make
  - PicoRuby compiler can't be integrated with mruby as it is
  - Should be "Rakenized"

Arts-Wallpapers.com

Rake

- Build process of mruby
  - libmruby_core.a
    - Includes mruby-compiler and murby's core objects implemented in C

Arts-Wallpapers.com

Rake

- Build process of mruby
  - libmruby_core.a
    - Includes mruby-compiler and murby's core objects implemented in C
  - mrbc
    - An executable of mruby-compiler

Arts-Wallpapers.com

Rake

- Build process of mruby
  - libmruby_core.a
    - Includes mruby-compiler and murby's core objects implemented in C
  - mrbc
    - An executable of mruby-compiler
  - mrblib.c
    - From `mrblib/*.rb`

Arts-Wallpapers.com

Rake

- Build process of mruby
  - libmruby_core.a
    - Includes mruby-compiler and murby's core objects implemented in C
  - mrbc
    - An executable of mruby-compiler
  - mrblib.c
    - From `mrblib/*.rb`
  - libmruby.a
    - Core objects, mrblib, mruby-compiler and mrbgems

Arts-Wallpapers.com

Rake

- Build process of mruby
  - libmruby_core.a
    - Includes mruby-compiler and murby's core objects implemented in C
  - mrbc
    - An executable of mruby-compiler
  - mrblib.c
    - From `mrblib/*.rb`
  - libmruby.a
    - Core objects, mrblib, mruby-compiler and mrbgems
  - mruby and mirb

Arts-Wallpapers.com

Rakenized PicoRuby

```
picoruby/
 ├──Rakefile    # modified `mruby/Rakefile`
 ├──bin/
 ├──build/
 ├──build_config/
 ├──include/
 ├──lib/        # cp from `mruby/lib`
 └──mrbgems/
      ├── mruby-bin-picoirb/
      ├── mruby-bin-picorbc/
      ├── mruby-bin-picoruby/
      ├── mruby-mrubyc/
      └── mruby-pico-compiler/
```

Arts-Wallpapers.com


SYNTAX

Syntax

- As of Episode I, PicoRuby has minimum syntax rules to implement PRK Firmware

Arts-Wallpapers.com

Syntax

- As of Episode I, PicoRuby has minimum syntax rules to implement PRK Firmware
- It's not enough to compile mrblib

Arts-Wallpapers.com

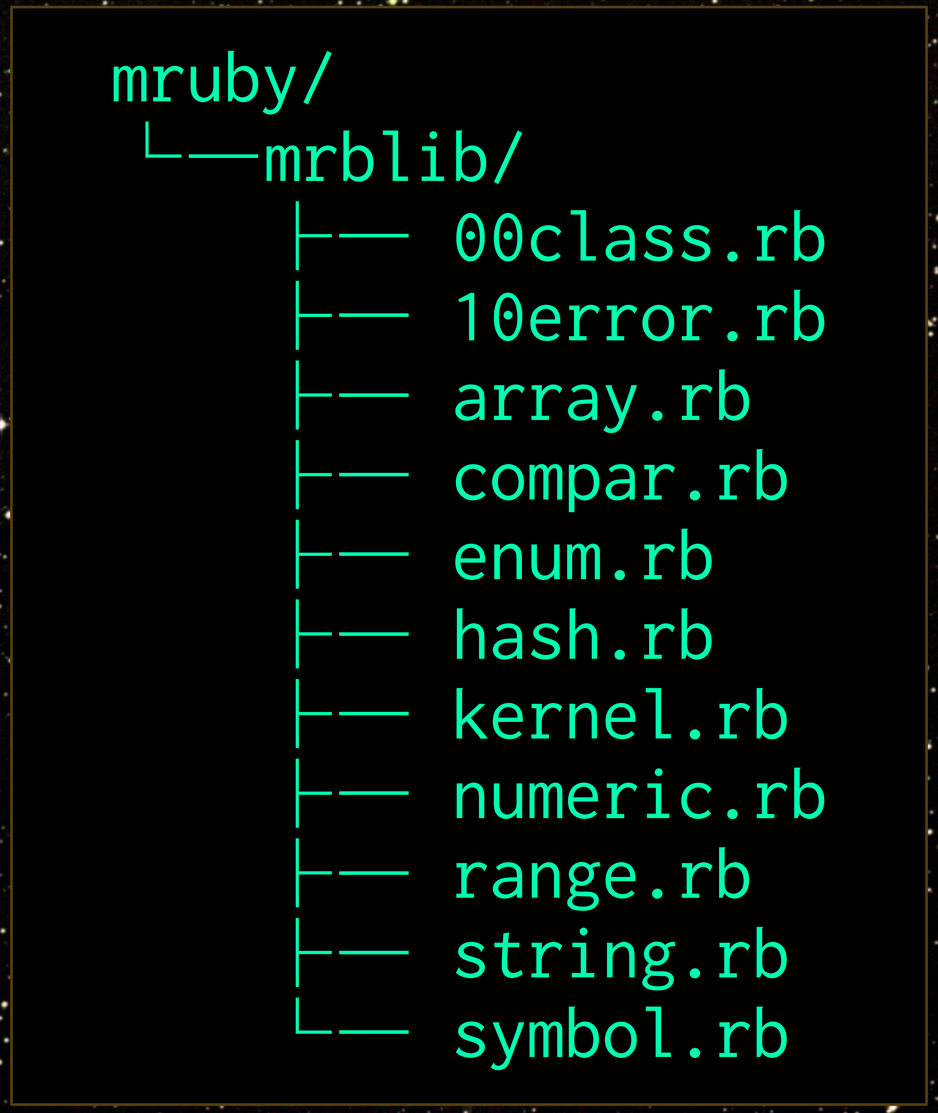
Syntax

- As of Episode I, PicoRuby has minimum syntax rules to implement PRK Firmware
- It's not enough to compile mrblib

mrblib?

Arts-Wallpapers.com

Syntax



Arts-Wallpapers.com

Syntax added

- Singleton method
- rescue-else-ensure-retry and rescue modifier
- Reswords (keyword as an identifier). eg) `next`
- Mass assignment
- Splat `*args` `**args`
- Assignable `p = p; p p #=> nil`
- `-2 ** 2 #=> -4`
- etc.

Arts-Wallpapers.com


PRESYM

Presym

- mruby 3.0 introduced **preallocated symbols** to save RAM
  - Presym is created in compilation time and located in ROM

Arts-Wallpapers.com

Presym

- mruby 3.0 introduced **preallocated symbols** to save RAM
  - Presym is created in compilation time and located in ROM
- From the compiler's point of view, presym is a described IREP

Arts-Wallpapers.com

Presym

- mruby 3.0 introduced **preallocated symbols** to save RAM
  - Presym is created in compilation time and located in ROM
- From the compiler's point of view, presym is a described IREP
- IREP: Internal representation
  - One IREP corresponds to a "Proc" in Ruby script

Arts-Wallpapers.com

Presym

```
class MyClass
  def my_method(ary = [])
    ary.each { |e| do_something }
  end
end
```

This Ruby script has four IREPs ("Procs") nested.

Arts-Wallpapers.com

Presym

```
#include <mruby.h>
#include <mruby/proc.h>
#include <mruby/presym.h>

#define mrb_BRACEÐ(...) {__VA_ARGS__}
#define mrb_ÐEFINE_SYMS_VAR(name, len, syms, qualifier) \
  static qualifier mrb_sym name[len] = mrb_BRACEÐ syms

static const mrb_code mrblib_proc_iseq_20[24] = {
0x35,0x04,0x00,0x00,0x14,0x03,0x01,0x04,0x01,0x43,0x03,0x27,0x03,0x00,0x05,0x16,0x03,0x25,0x00,0x02,0x15,0x03,0x39,0x03,};
mrb_ÐEFINE_SYMS_VAR(mrblib_proc_lv_20, 2, (MRB_SYM(other), MRB_OPSYM(and), ), const);
static const mrb_irep mrblib_proc_irep_20 = {
  3,6,0,
  MRB_IREP_STATIC,mrblib_proc_iseq_20,
  NULL,NULL,NULL,
  mrblib_proc_lv_20,
  NULL,                                 /* debug_info */
  24,0,0,0,0
};
static const mrb_irep *mrblib_proc_reps_1[1] = {
  &mrblib_proc_irep_20,
};
mrb_ÐEFINE_SYMS_VAR(mrblib_proc_syms_1, 1, (MRB_OPSYM(neq), ), const);
static const mrb_code mrblib_proc_iseq_1[13] = {
0x68,0x01,0x5b,0x02,0x00,0x64,0x01,0x00,0x11,0x01,0x00,0x39,0x01,};
static const mrb_irep mrblib_proc_irep_1 = {
  1,3,0,
  MRB_IREP_STATIC,mrblib_proc_iseq_1,
  NULL,mrblib_proc_syms_1,mrblib_proc_reps_1,
  NULL,                                 /* lv */
  NULL,                                 /* debug_info */
  13,0,1,1,0
};
// ... goes on
```

Arts-Wallpapers.com

Presym

```c
/* Program data array struct */
typedef struct mrb_irep {
  uint16_t nlocals;         /* Number of local variables */
  uint16_t nregs;           /* Number of register variables */
  uint16_t clen;            /* Number of catch handlers */
  uint8_t flags;

  const mrb_code *iseq;
  /*
  * A catch handler table is placed after the iseq entity.
  * The reason it doesn't add fields to the structure is to keep the mrb_irep structure from bloating.
  * The catch handler table can be obtained with `mrb_irep_catch_handler_table(irep)`.
  */
  const mrb_pool_value *pool;
  const mrb_sym *syms;
  const struct mrb_irep * const *reps;

  const mrb_sym *lv;
  /* debug info */
  struct mrb_irep_debug_info* debug_info;

  uint32_t ilen;
  uint16_t plen, slen;
  uint16_t rlen;
  uint16_t refcnt;
} mrb_irep;
```

Arts-Wallpapers.com

Presym

```
typedef struct scope {
  uint32_t irep_parameters; /* bbb */
  uint32_t nest_stack; /* Initial: 00000000 00000000 00000000 00000001 */
  Scope *upper;
  Scope *first_lower;
  Scope *next;
  bool lvar_top;
  uint16_t next_lower_number;
  uint16_t nlowers;  /* irep.rlen in mruby/c (num of child IREP block) */
  CodePool *first_code_pool;
  CodePool *current_code_pool;
  unsigned int nlocals;
  Symbol *symbol;
  Lvar *lvar;
  Literal *literal;
  GenLiteral *gen_literal; /* Exceptional literals in generator */
  unsigned int sp;
  uint32_t ilen; /* irep length */
  uint16_t slen; /* symbol length */
  uint16_t plen; /* pool length */
  uint16_t clen; /* exception handler length */
  uint16_t vm_code_size;
  uint8_t *vm_code;
  BreakStack *break_stack;
  RetryStack *retry_stack;
  AssignSymbol *last_assign_symbol;
  Backpatch *backpatch; /* for backpatching of JMP label */
  int nargs_before_splat;
  uint8_t gen_splat_status;
  uint8_t gen_array_status;
  uint8_t gen_array_count;
  ExcHandler *exc_handler;
} Scope;
```

Arts-Wallpapers.com


MICRORUBY

MicroRuby

```ruby
# mruby/build_config/microruby.rb
MRuby::Build.new do |conf|
  conf.toolchain
  conf.mrbcfile = "#{conf.build_dir}/bin/picorbc"
  conf.gem git: "https://github.com/hasumikin/mruby-bin-picorbc.git", branch: "master"
  conf.gem git: "https://github.com/hasumikin/mruby-pico-compiler.git", branch: "master"
  conf.gem git: "https://github.com/hasumikin/mruby-bin-microruby.git", branch: "master"
  conf.gem core: "mruby-print"
  conf.cc.defines << "MICRORUBY"
  conf.cc.defines << "NDEBUG"
  conf.cc.defines << "MRBC_ALLOC_LIBC"
  conf.cc.defines << "REGEX_USE_ALLOC_LIBC"
end
```

```
cd mruby && git checkout 3.0.0
MRUBY_CONFIG=microruby rake
```

Arts-Wallpapers.com

MicroRuby

- Build process of MicroRuby
  - libmruby_core.a
    - Doesn't include an mruby-related object

Arts-Wallpapers.com

MicroRuby

- Build process of MicroRuby
  - libmruby_core.a
    - Doesn't include an mruby-related object
  - picorbc
    - An executable of picoruby-compiler

Arts-Wallpapers.com

MicroRuby

- Build process of MicroRuby
  - libmruby_core.a
    - Doesn't include an mruby-related object
  - picorbc
    - An executable of picoruby-compiler
  - mrblib.c
    - From `mrblib/*.rb` compiled by picorbc

Arts-Wallpapers.com

MicroRuby

- Build process of MicroRuby
  - libmruby_core.a
    - Doesn't include an mruby-related object
  - picorbc
    - An executable of picoruby-compiler
  - mrblib.c
    - From `mrblib/*.rb` compiled by picorbc
  - libmruby.a
    - Core objects of mruby, mrblib and picoruby-compiler

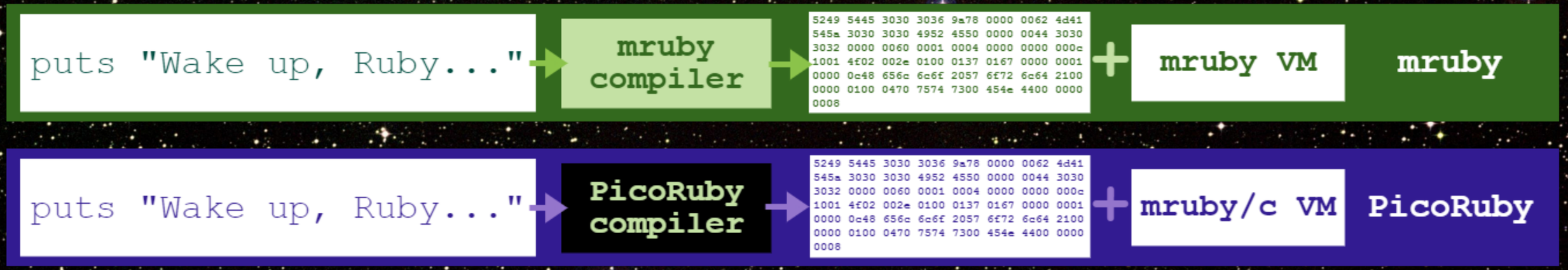
Arts-Wallpapers.com

MicroRuby

- Build process of MicroRuby
  - libmruby_core.a
    - Doesn't include an mruby-related object
  - picorbc
    - An executable of picoruby-compiler
  - mrblib.c
    - From `mrblib/*.rb` compiled by picorbc
  - libmruby.a
    - Core objects of mruby, mrblib and picoruby-compiler
  - microruby executable
    - microirb, not yet

Arts-Wallpapers.com

MicroRuby



Arts-Wallpapers.com

MicroRuby



Arts-Wallpapers.com



Arts-Wallpapers.com

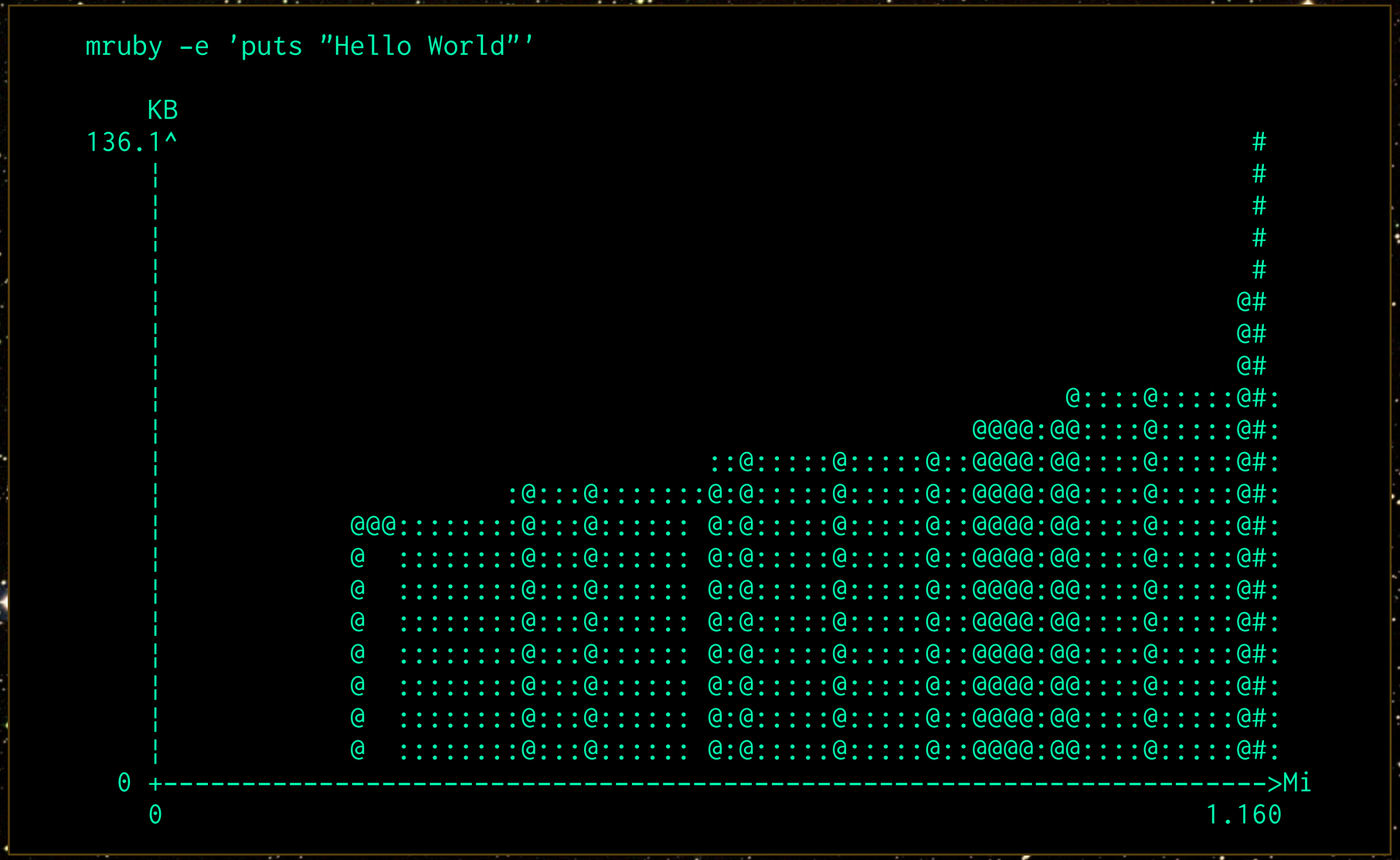
massif

```
$ valgrind                                          \
    --tool=massif                                   \
    --stacks=yes                                    \
    path/to/(mruby¦microrby¦picoruby)               \
    -e 'puts "Hello World"'

$ ms_print massif.out.xxx ¦ less
```

Arts-Wallpapers.com

massif

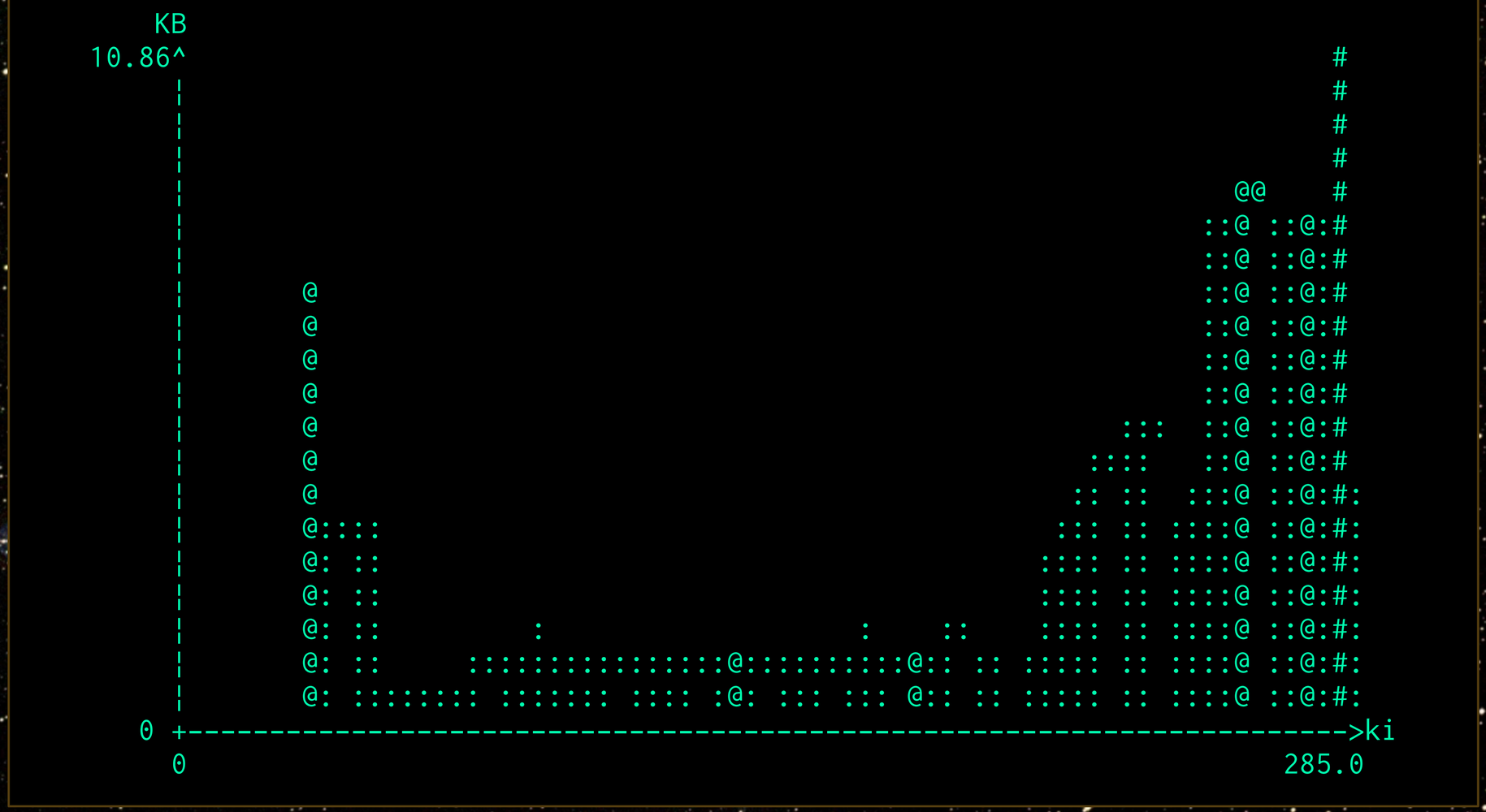


massif



massif

```
picoruby -e 'puts "Hello World"'
```



massif

```
# (mruby¦microruby¦picoruby) \
#  -e 'puts "Hello World"'
-------------------------------
   mruby      MicroRuby    PicoRuby
===============================
  136.1 KB    88.05 KB      10.86 KB
-------------------------------
```

(Note)

All the figures on 64-bit architecture with glibc

They're probably 25-35% smaller on 32-bit

Arts-Wallpapers.com


EPISODE III?

Episode III?

- Struct "mrb_context" now bothers me
  - Holds information like local variables so to IRB work

Arts-Wallpapers.com

Episode III?

- Struct "mrb_context" now bothers me
  - Holds information like local variables so to IRB work
  - Hard to handle due to intricate dependency among mrb_state, mrb_context and mrb_irep

Arts-Wallpapers.com

Episode III?

- Struct "mrb_context" now bothers me
  - Holds information like local variables so to IRB work
  - Hard to handle due to intricate dependency among mrb_state, mrb_context and mrb_irep
  - PicoRuby compiler doesn't want to include mruby's header files

Arts-Wallpapers.com

Episode III?

- Struct "mrb_context" now bothers me
  - Holds information like local variables so to IRB work
  - Hard to handle due to intricate dependency among mrb_state, mrb_context and mrb_irep
  - PicoRuby compiler doesn't want to include mruby's header files
- Possible solutions:
  1. Fixing mruby
  2. Fixing PicoRuby compiler to make an IREP that can be cast into mrb_irep

Arts-Wallpapers.com

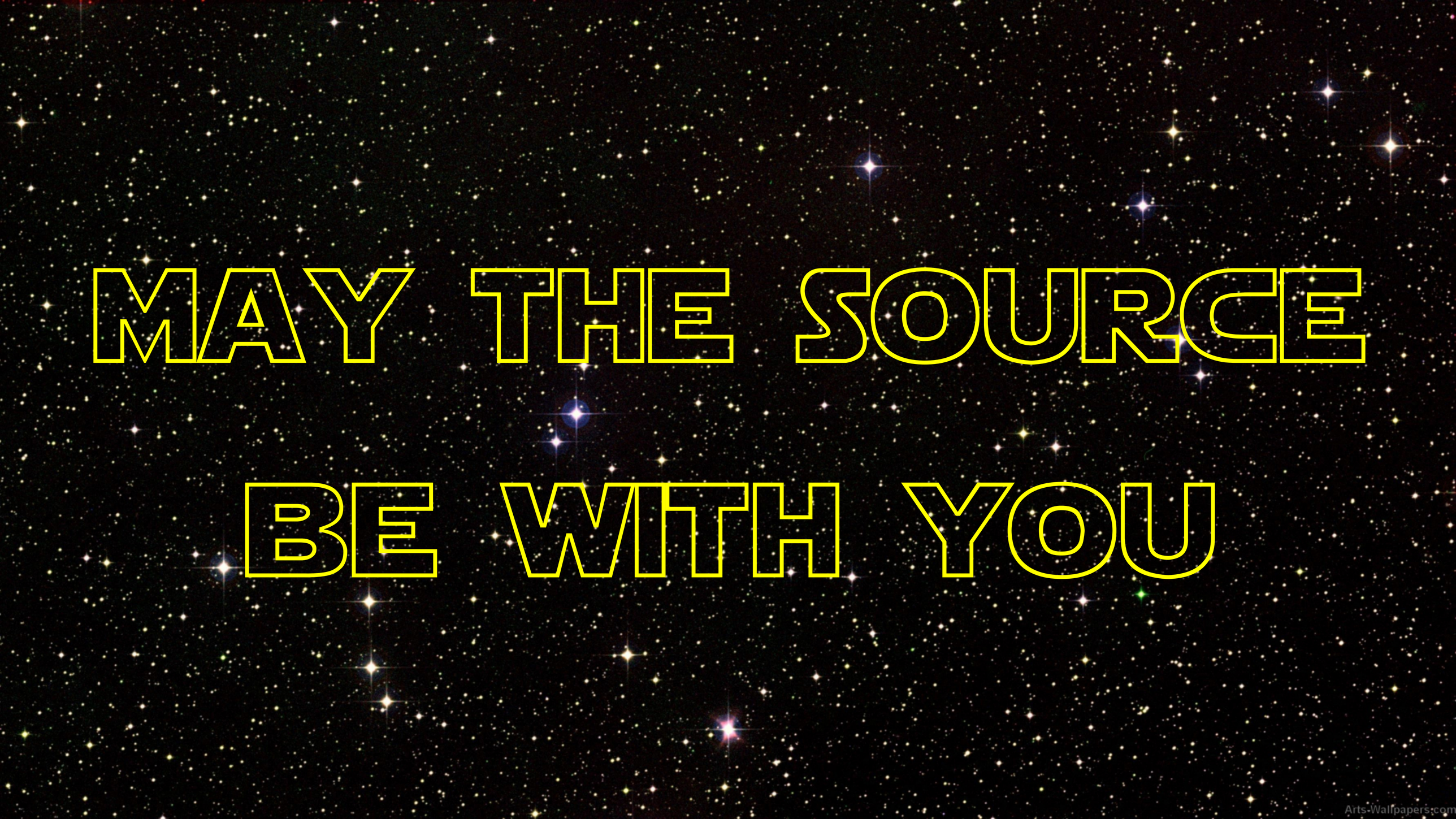
THE IREP STRIKES BACK

Arts-Wallpapers.com

WE'RE HIRING!

Monstarlab ⁚⁚

Arts-Wallpapers.com



Arts-Wallpapers.com