Source Code Generators

fundamentals

https://github.com/jkubin/if19

InstallFest 2019

Josef Kubín

Contents

- Examples in M4 and XSLT, (de)motivation
- Formal grammar (Chomsky)
  - Context-Free Grammar
- Automata (abstract machines)
- Buffers (Java example, proverbs)
- Workflow
  - Vim abbreviations & placeholders
  - Debugging
- Short history
- Examples, Q&A

Motivation plus

- Interdependent source code
  - Consistency
- Difficult source code
  - Easy to change
- Utilization of skilled people
  - Unskilled people run code generator
- Single man project
  - Knowledge is encoded in the code generator
- Fast response to customer requirements
- Exact output
- No bugs!!!

Motivation minus

- Not as easy as it looks …
- An unusual programming paradigm
- Heavily depends on developer experience
  - Necessary theoretical computer science background
- Slower development (it must finally pay off)
  - At least 2x slower for unskilled people
- Maintenance (could be hard)
  - Badly written code is not readable or writable later
    - https://en.wikipedia.org/wiki/Big_ball_of_mud
- Free syntax

```
$ m4 master.m4 -DSYM=a,b,c script.m4 config.mc > target…
```

config.mc

Source Code Generators
part I (today)

```
CAPT([…])
ITEM([…], […])
XYZ([…], […], […], […], […],
[…], […], […], …)
```

- Simple records
  Single-language
  Could be problematic

Source Code Generators
part II (next lecture)

- Multiple-records
  Multiple-languages
  All in one place

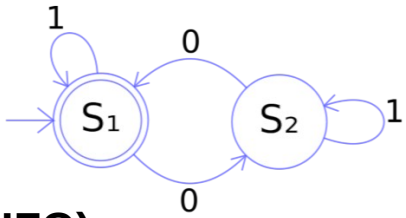
```
CAPT([HARDWARE])
ITEM([MCU_WATCHDOG])
XYZ([ERR_MESSAGE])
```

script.m4

1) Context-Free Grammar (CFG)
Production rules
P: **A** → **β**
A ∈ **N**
β ∈ **(N** ∪ **Σ)***

M4 Production rules
P: define([**A**], [**β**])

2) Automata



3) Buffers (FIFO)
divert(-1) → */dev/null*
divert(0) → *stdout*
divert(1) → #1.BFR
divert(2) → #2.BFR
…
undivert(1) #1.BFR → ...
undivert(2) #2.BFR → ...
...

4) Stacks
pushdef([**A**], [**β**]) popdef([**A**])

target…

- **csv**
- **ini**
- **c**
- **h**
- **XML**
- **HTML**
- **JSON**
- **Java**
- **Bash**

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

```
installfest_2019.mc          1,1          Top
```

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#           _____CAPT
#        |     /
#        __V___/
# --->/ CAPT \
#     \______/
#
#           _____ITEM
#        |     /
#        __V___/
# --->/ ITEM \
#     \______/
#

# A → β
define([CAPT], [

        define([SECTION_NAME], [$1])
])

# A → β
define([ITEM], [

        divert(0)dnl
SECTION_NAME[   $1      $2]
divert(-1)
])
~
~
~
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
__HEADER([Josef Kubin], [2019/02/21], [embedded_dia»

#           _____CAPT
#          |    /
#          __V___/
# --->/ CAPT \
#      \______/
#
#           _____ITEM
#          |    /
#          __V___/
# --->/ ITEM \
#      \______/
#

# A → β
define([CAPT], [

        divert(0)dnl

[[$1]]
divert(-1)
])

# A → β
define([ITEM], [

        divert(0)dnl
[$1=$2]
divert(-1)
])

divert(0)dnl
; DO_NOT_EDIT
divert(-1)
~
~
~
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#          ______
# --->/ CAPT \---,CAPT
#     \______/<--'
#          ______
# --->/ ITEM \---,ITEM
#     \______/<--'
#
#              _____CAPT_____
#            |               |
#  CAPT   _V_   ITEM    ____|_____
# ----->/ ε \------>/ [\n"LF"] \---,ITEM
#       \___/       \__________/<--'
# β
define([EPSILON_FIRST], [define([$0], [\n"
"])])

# A → β
define([CAPT], [

        divert(0)dnl
undivert
char PROJECT[_$1][[]] =
"divert(1)";
divert(-1)

        # A → β
        define([NEW_LINE], defn([EPSILON_FIRST]))
])

# A → β
define([ITEM], [

        divert(0)NEW_LINE[$2]dnl
divert(-1)
])

divert(0)dnl
/*
 * DO_NOT_EDIT
 */
divert(-1)
~
~
~
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

```
installfest_2019.mc          1,1          Top
```

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#           _____CAPT
#          |    /
#          __V___/
# --->/ CAPT \
#     \______/
#
#           _____ITEM
#          |    /
#          __V___/
# --->/ ITEM \
#     \______/
#

# A → β
define([CAPT], [

        divert(0)dnl
undivert
char *PROJECT[_$1][[]] = {
divert(1)dnl
};
divert(-1)
])

# A → β
define([ITEM], [

        divert(0)dnl
        "[$2]",
divert(-1)
])

divert(0)dnl
/*
 * DO_NOT_EDIT
 */
divert(-1)
~
~
~
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#           _____CAPT
#          |    /
#          __V___/
# --->/ CAPT \
#     \______/
#
#           _____ITEM
#          |    /
#          __V___/
# --->/ ITEM \
#     \______/
#

# A → β
define([CAPT], [

        define([CAPT_NAME], translit([$1], [a-z], [»

        divert(0)
/* PROJECT[_$1] */
divert(-1)

        # init or reset counter
        define([COUNTER], [1])
])

# A → β
define([ITEM], [

        divert(0)dnl
[#]define CAPT_NAME[_$1]        0x[]eval(COUNTER, 1»
divert(-1)

        # counter++
        define([COUNTER], incr(COUNTER))
])

divert(0)dnl
/*
 * DO_NOT_EDIT
 */
#ifndef __LOG_INDEX_FROM_MACROS_H
#define __LOG_INDEX_FROM_MACROS_H
divert(2)
#endif /* __LOG_INDEX_FROM_MACROS_H */
divert(-1)
~
~
~
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

```
installfest_2019.mc          1,1          Top
```

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#          _____
# --->/ CAPT \---,CAPT
#     \_____/<--'
#          _____
# --->/ ITEM \---,ITEM
#     \_____/<--'
#
#                 _____CAPT_______
#               |                |
#  CAPT  _____V_____  ITEM   _|_
# ----->/ [= COUNTER] \------>/ ε \---,ITEM
#       \____________/       \___/<--'
# β
define([PRINT_ONCE], [ = COUNTER[]define([$0])])

# A → β
define([CAPT], [

        # A → β
        define([START_VAL], defn([PRINT_ONCE]))

        define([CAPT_NAME], translit([$1], [a-z], [»

        divert(0)dnl
undivert(1)
enum $1_log {
divert(1)dnl
};
divert(-1)

        # counter = 1
        define([COUNTER], [1])
])

# A → β
define([ITEM], [

        divert(0)dnl
        CAPT_NAME[_$1]START_VAL,        /* 0x[]eval»
divert(-1)

        # counter++
        define([COUNTER], incr(COUNTER))
])

divert(0)dnl
/*
 * DO_NOT_EDIT
 */
#ifndef __LOG_INDEX_FROM_ENUM_H
#define __LOG_INDEX_FROM_ENUM_H
divert(2)
#endif /* __LOG_INDEX_FROM_ENUM_H */
divert(-1)
~
~
~
~
~
```

```
h_enum.h.m4                 12,2          All
```

```
/*
 * DO NOT EDIT! Generated automatically!
 */
#ifndef __LOG_INDEX_FROM_ENUM_H
#define __LOG_INDEX_FROM_ENUM_H

enum hardware_log {
        HARDWARE_AD_OFFLINE = 1,        /* 0x01 "AD»
        HARDWARE_AD_WATCHDOG,   /* 0x02 "AD watchdo»
        HARDWARE_RTC_OFFLINE,   /* 0x03 "RTC offlin»
        HARDWARE_RTC_NO_SQW,    /* 0x04 "no SQW to »
        HARDWARE_MCU_WDT,       /* 0x05 "MCU watchd»
        HARDWARE_ETH_TRANSC_OFFLINE,    /* 0x06 "ET»
        HARDWARE_ETH_BUS_ERROR, /* 0x07 "ETH bus in»
};

enum memory_log {
        MEMORY_DFLASH_OFFLINE = 1,      /* 0x01 "Da»
        MEMORY_DFLASH_ERASE_WRITE,      /* 0x02 "Da»
        MEMORY_FRAM_OFFLINE,    /* 0x03 "FRAM offli»
        MEMORY_FRAM_WRITE,      /* 0x04 "FRAM can n»
        MEMORY_SRAM_OFFLINE,    /* 0x05 "SRAM offli»
        MEMORY_SRAM_WRITE,      /* 0x06 "SRAM can n»
};

enum conf_log {
        CONF_AD_CONFIG = 1,     /* 0x01 "AD config"»
        CONF_MAIN_CONFIG,       /* 0x02 "main confi»
        CONF_UART_A,    /* 0x03 "UART A: changed ad»
        CONF_UART_B,    /* 0x04 "UART B: changed ad»
        CONF_UART_C,    /* 0x05 "UART C: changed ad»
        CONF_UART_D,    /* 0x06 "UART D: changed ad»
};

enum proto_log {
        PROTO_MBUS_A = 1,       /* 0x01 "M-Bus A: s»
        PROTO_MBUS_B,   /* 0x02 "M-Bus B: settings »
        PROTO_MBUS_C,   /* 0x03 "M-Bus C: settings »
        PROTO_MBUS_D,   /* 0x04 "M-Bus D: settings »
        PROTO_MODBUS_A, /* 0x05 "MODBUS A: settings»
        PROTO_MODBUS_B, /* 0x06 "MODBUS B: settings»
        PROTO_MODBUS_C, /* 0x07 "MODBUS C: settings»
        PROTO_MODBUS_D, /* 0x08 "MODBUS D: settings»
};

enum usr_log {
        USR_MESSAGE = 1,        /* 0x01 "změna konf»
        USR_MAIN_CONFIG,        /* 0x02 "hlavní kon»
        USR_MSG_FLAGS,  /* 0x03 "vymazání chybových»
        USR_USE_PASSWD, /* 0x04 "uživatelské heslo"»
        USR_USR_CONST,  /* 0x05 "uživatelské konsta»
        USR_LAST_KNOWN_TIME,    /* 0x06 "nastaven p»
        USR_NEW_TIME,   /* 0x07 "nově nastavený čas»
        USR_PREV_TIME,  /* 0x08 "předchozí čas" */
        USR_SUMMER_TO_WINTER,   /* 0x09 "změna času»
        USR_WINTER_TO_SUMMER,   /* 0x0a "změna času»
};

enum on_off_log {
        ON_OFF_START_TIME = 1,  /* 0x01 "čas zapnut»
        ON_OFF_SHUTDOWN_TIME,   /* 0x02 "čas vypnut»
```

```
h_enum.h                    1,1           Top
:q
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#           _____CAPT
#          |    /
#          __V___/
# --->/ CAPT \
#     \______/
#
#           _____ITEM
#          |    /
#          __V___/
# --->/ ITEM \
#     \______/
#

# A → β
define([CAPT], [

        divert(0)dnl
undivert(1)
struct $1_log {
divert(1)dnl
};

divert(2)dnl
[#]define translit([$1], [a-z], [A-Z])(code)    (un»
divert(-1)

        # counter = 0
        define([COUNTER], [0])
])

# β
define([ITEM], [

        divert(0)dnl
        /* 0x[]eval(COUNTER, 16, 2) */ unsigned cha»
divert(-1)

        # counter++
        define([COUNTER], incr(COUNTER))
])

divert(0)dnl
/*
 * DO_NOT_EDIT
 */
#ifndef __LOG_INDEX_FROM_STRUCT_H
#define __LOG_INDEX_FROM_STRUCT_H
divert(3)
#endif /* __LOG_INDEX_FROM_STRUCT_H */
divert(-1)
~
~
~
~
~
~
~
~
~
```

h_structs.h.m4 48,2 All

```
/*
 * DO NOT EDIT! Generated automatically!
 */
#ifndef __LOG_INDEX_FROM_STRUCT_H
#define __LOG_INDEX_FROM_STRUCT_H

struct hardware_log {
        /* 0x00 */ unsigned char AD_OFFLINE;     /* »
        /* 0x01 */ unsigned char AD_WATCHDOG;    /* »
        /* 0x02 */ unsigned char RTC_OFFLINE;    /* »
        /* 0x03 */ unsigned char RTC_NO_SQW;     /* »
        /* 0x04 */ unsigned char MCU_WDT;        /* »
        /* 0x05 */ unsigned char ETH_TRANSC_OFFLINE»
        /* 0x06 */ unsigned char ETH_BUS_ERROR; /* »
};


struct memory_log {
        /* 0x00 */ unsigned char DFLASH_OFFLINE;    »
        /* 0x01 */ unsigned char DFLASH_ERASE_WRITE»
        /* 0x02 */ unsigned char FRAM_OFFLINE;  /* »
        /* 0x03 */ unsigned char FRAM_WRITE;    /* »
        /* 0x04 */ unsigned char SRAM_OFFLINE;  /* »
        /* 0x05 */ unsigned char SRAM_WRITE;    /* »
};


struct conf_log {
        /* 0x00 */ unsigned char AD_CONFIG;      /* »
        /* 0x01 */ unsigned char MAIN_CONFIG;    /* »
        /* 0x02 */ unsigned char UART_A;         /* »
        /* 0x03 */ unsigned char UART_B;         /* »
        /* 0x04 */ unsigned char UART_C;         /* »
        /* 0x05 */ unsigned char UART_D;         /* »
};


struct proto_log {
        /* 0x00 */ unsigned char MBUS_A;         /* »
        /* 0x01 */ unsigned char MBUS_B;         /* »
        /* 0x02 */ unsigned char MBUS_C;         /* »
        /* 0x03 */ unsigned char MBUS_D;         /* »
        /* 0x04 */ unsigned char MODBUS_A;       /* »
        /* 0x05 */ unsigned char MODBUS_B;       /* »
        /* 0x06 */ unsigned char MODBUS_C;       /* »
        /* 0x07 */ unsigned char MODBUS_D;       /* »
};


struct usr_log {
        /* 0x00 */ unsigned char MESSAGE;        /* »
        /* 0x01 */ unsigned char MAIN_CONFIG;    /* »
        /* 0x02 */ unsigned char MSG_FLAGS;      /* »
        /* 0x03 */ unsigned char USE_PASSWD;     /* »
        /* 0x04 */ unsigned char USR_CONST;      /* »
        /* 0x05 */ unsigned char LAST_KNOWN_TIME;  »
        /* 0x06 */ unsigned char NEW_TIME;       /* »
        /* 0x07 */ unsigned char PREV_TIME;      /* »
        /* 0x08 */ unsigned char SUMMER_TO_WINTER; »
        /* 0x09 */ unsigned char WINTER_TO_SUMMER; »
};
```

h_structs.h 1,1 Top

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
__HEADER([Josef Kubin], [2019/02/17], [embedded dia»

#           _____CAPT
#         |     /
#         __V___/
# --->/ CAPT \
#     \______/
#
#           _____ITEM
#         |     /
#         __V___/
# --->/ ITEM \
#     \______/
#

# A → β
define([CAPT], [

        divert(0)dnl
undivert(1)
struct $1_log {
divert(1)dnl
};

divert(2)dnl
[#]define translit([$1], [a-z], [A-Z])(code)    (un»
divert(-1)

        # counter = 0
        define([COUNTER], [0])
])

# β
define([ITEM], [

        divert(0)dnl
        /* 0x[]eval(COUNTER, 16, 2) */ unsigned cha»
divert(-1)

        # counter++
        define([COUNTER], incr(COUNTER))
])

divert(0)dnl
/*
 * DO_NOT_EDIT
 */
#ifndef __LOG_INDEX_FROM_STRUCT_H
#define __LOG_INDEX_FROM_STRUCT_H
divert(3)
#endif /* __LOG_INDEX_FROM_STRUCT_H */
divert(-1)
~
~
~
~
~
~
~
~
~
```

h_structs.h.m4 48,2 All

```
};


struct usr_log {
        /* 0x00 */ unsigned char MESSAGE;       /* »
        /* 0x01 */ unsigned char MAIN_CONFIG;   /* »
        /* 0x02 */ unsigned char MSG_FLAGS;     /* »
        /* 0x03 */ unsigned char USE_PASSWD;    /* »
        /* 0x04 */ unsigned char USR_CONST;     /* »
        /* 0x05 */ unsigned char LAST_KNOWN_TIME;  »
        /* 0x06 */ unsigned char NEW_TIME;      /* »
        /* 0x07 */ unsigned char PREV_TIME;     /* »
        /* 0x08 */ unsigned char SUMMER_TO_WINTER; »
        /* 0x09 */ unsigned char WINTER_TO_SUMMER; »
};


struct on_off_log {
        /* 0x00 */ unsigned char START_TIME;    /* »
        /* 0x01 */ unsigned char SHUTDOWN_TIME; /* »
};

#define HARDWARE(code)  (unsigned char)&((struct ha»
#define MEMORY(code)    (unsigned char)&((struct me»
#define CONF(code)      (unsigned char)&((struct co»
#define PROTO(code)     (unsigned char)&((struct pr»
#define USR(code)       (unsigned char)&((struct us»
#define ON_OFF(code)    (unsigned char)&((struct on»

#endif /*   LOG_INDEX_FROM_STRUCT_H */
```

h_structs.h 76,1 Bot

```
nsigned char MESSAGE;       /* "změna konfigurace c»
nsigned char MAIN_CONFIG;   /* "hlavní konfigurace"»
nsigned char MSG_FLAGS;     /* "vymazání chybových »
nsigned char USE_PASSWD;    /* "uživatelské heslo" »
nsigned char USR_CONST;     /* "uživatelské konstan»
nsigned char LAST_KNOWN_TIME;       /* "nastaven po»
nsigned char NEW_TIME;      /* "nově nastavený čas"»
nsigned char PREV_TIME;     /* "předchozí čas" */
nsigned char SUMMER_TO_WINTER;      /* "změna času:»
nsigned char WINTER_TO_SUMMER;      /* "změna času:»



nsigned char START_TIME;    /* "čas zapnutí" */
nsigned char SHUTDOWN_TIME; /* "čas vypnutí" */


e)  (unsigned char)&((struct hardware_log *)0)->code
    (unsigned char)&((struct memory_log *)0)->code
    (unsigned char)&((struct conf_log *)0)->code
    (unsigned char)&((struct proto_log *)0)->code
    (unsigned char)&((struct usr_log *)0)->code
    (unsigned char)&((struct on_off_log *)0)->code


X_FROM_STRUCT_H */
```

h_structs.h 69,71-72 Bot

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#            _____CAPT
#        |    /
#        __V___/
# --->/ CAPT \
#     \______/
#
#            _____ITEM
#        |    /
#        __V___/
# --->/ ITEM \
#     \______/
#

# A → β
define([CAPT], [

        divert(0)dnl
undivert(1)dnl
        [<$1>]
divert(1)dnl
        [</$1>]
divert(-1)
])

# A → β
define([ITEM], [

        divert(0)dnl
                [<item><name>$1</name><value>$2</va»
divert(-1)
])

divert(0)dnl
<!-- DO_NOT_EDIT -->
<?xml version="1.0" encoding="utf-8"?>
<PROJECT>
divert(2)dnl
</PROJECT>
divert(-1)
~
~
~
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
# vim: set ft=m4:
# ^^^^ this is Vim modeline (to force M4 filetype)
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#            _____CAPT
#           |    /
#           __V___/
# --->/ CAPT \
#     \_____/
#
#            _____ITEM
#           |    /
#           __V___/
# --->/ ITEM \
#     \_____/
#

# A → β
define([CAPT], [

        divert(0)dnl
undivert(1)dnl
        <section>
                <h2>[$1]</h2>
                <ol>
divert(1)dnl
                </ol>
        </section>
divert(-1)
])

# A → β
define([ITEM], [

        divert(0)dnl
                        [<li><b>$1</b>: $2</li>]
divert(-1)
])

define([TITLE], [Power of M4])

divert(0)dnl
<!DOCTYPE html>
<!-- DO_NOT_EDIT -->
<html>
<head>
<meta charset="UTF-8">
<title>TITLE</title>
</head>
<body>
<header>
<h1>TITLE</h1>
</header>
<article>
divert(2)dnl
</article>
<footer>
        <p><a href="PROJECT_ROOT_URL/PROJECT">PROJE»
</footer>
</body>
</html>
```

html_simple.html.m4 1,1 Top

```
<!DOCTYPE html>
<!-- DO NOT EDIT! Generated automatically! -->
<html>
<head>
<meta charset="UTF-8">
<title>Power of M4</title>
</head>
<body>
<header>
<h1>Power of M4</h1>
</header>
<article>
        <section>
                <h2>hardware</h2>
                <ol>
                        <li><b>AD_OFFLINE</b>: AD o»
                        <li><b>AD_WATCHDOG</b>: AD »
                        <li><b>RTC_OFFLINE</b>: RTC»
                        <li><b>RTC_NO_SQW</b>: no S»
                        <li><b>MCU_WDT</b>: MCU wat»
                        <li><b>ETH_TRANSC_OFFLINE</»
                        <li><b>ETH_BUS_ERROR</b>: E»
                </ol>
        </section>
        <section>
                <h2>memory</h2>
                <ol>
                        <li><b>DFLASH_OFFLINE</b>: »
                        <li><b>DFLASH_ERASE_WRITE</»
                        <li><b>FRAM_OFFLINE</b>: FR»
                        <li><b>FRAM_WRITE</b>: FRAM»
                        <li><b>SRAM_OFFLINE</b>: SR»
                        <li><b>SRAM_WRITE</b>: SRAM»
                </ol>
        </section>
        <section>
                <h2>conf</h2>
                <ol>
                        <li><b>AD_CONFIG</b>: AD co»
                        <li><b>MAIN_CONFIG</b>: mai»
                        <li><b>UART_A</b>: UART A: »
                        <li><b>UART_B</b>: UART B: »
                        <li><b>UART_C</b>: UART C: »
                        <li><b>UART_D</b>: UART D: »
                </ol>
        </section>
        <section>
                <h2>proto</h2>
                <ol>
                        <li><b>MBUS_A</b>: M-Bus A:»
                        <li><b>MBUS_B</b>: M-Bus B:»
                        <li><b>MBUS_C</b>: M-Bus C:»
                        <li><b>MBUS_D</b>: M-Bus D:»
                        <li><b>MODBUS_A</b>: MODBUS»
                        <li><b>MODBUS_B</b>: MODBUS»
                        <li><b>MODBUS_C</b>: MODBUS»
                        <li><b>MODBUS_D</b>: MODBUS»
                </ol>
        </section>
        <section>
                <h2>usr</h2>
```

html_simple.html 1,1 Top

"html_simple.html" 87L, 2783C

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
# vim: set ft=m4:
# ^^^^ this is Vim modeline (to force M4 filetype)
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#            _____CAPT
#        |    /
#        __V___/
# --->/ CAPT \
#     \______/
#
#            _____ITEM
#        |    /
#        __V___/
# --->/ ITEM \
#     \______/
#

# A → β
define([CAPT], [

        divert(0)dnl
                [<li><a href="#$1">$1</a></li>]
divert(2)dnl
undivert(3)dnl
        <section>
                [<h2 id="$1">$1</h2>]
                <ol>
divert(3)dnl
                </ol>
        </section>
divert(-1)
])

# A → β
define([ITEM], [

        divert(2)dnl
                        [<li><b>$1</b>: $2</li>]
divert(-1)
])

define([TITLE], [Power of M4])

divert(0)dnl
<!DOCTYPE html>
<!-- DO_NOT_EDIT -->
<html>
<head>
<meta charset="UTF-8">
<title>TITLE</title>
<style>
html {
        font-size: 120%;
}
section ul li {
        display: inline;
}
</style>
</head>
<body>
<header>
```

html_navigation.html.m4 2,1 Top

```
<!DOCTYPE html>
<!-- DO NOT EDIT! Generated automatically! -->
<html>
<head>
<meta charset="UTF-8">
<title>Power of M4</title>
<style>
html {
        font-size: 120%;
}
section ul li {
        display: inline;
}
</style>
</head>
<body>
<header>
<h1>Power of M4</h1>
</header>
<nav>
        <ul>
                <li><a href="#hardware">hardware</a»
                <li><a href="#memory">memory</a></l»
                <li><a href="#conf">conf</a></li>
                <li><a href="#proto">proto</a></li>
                <li><a href="#usr">usr</a></li>
                <li><a href="#on_off">on_off</a></l»
        </ul>
</nav>
<article>
        <section>
                <h2 id="hardware">hardware</h2>
                <ol>
                        <li><b>AD_OFFLINE</b>: AD o»
                        <li><b>AD_WATCHDOG</b>: AD »
                        <li><b>RTC_OFFLINE</b>: RTC»
                        <li><b>RTC_NO_SQW</b>: no S»
                        <li><b>MCU_WDT</b>: MCU wat»
                        <li><b>ETH_TRANSC_OFFLINE</»
                        <li><b>ETH_BUS_ERROR</b>: E»
                </ol>
        </section>
        <section>
                <h2 id="memory">memory</h2>
                <ol>
                        <li><b>DFLASH_OFFLINE</b>: »
                        <li><b>DFLASH_ERASE_WRITE</»
                        <li><b>FRAM_OFFLINE</b>: FR»
                        <li><b>FRAM_WRITE</b>: FRAM»
                        <li><b>SRAM_OFFLINE</b>: SR»
                        <li><b>SRAM_WRITE</b>: SRAM»
                </ol>
        </section>
        <section>
                <h2 id="conf">conf</h2>
                <ol>
                        <li><b>AD_CONFIG</b>: AD co»
                        <li><b>MAIN_CONFIG</b>: mai»
                        <li><b>UART_A</b>: UART A: »
                        <li><b>UART_B</b>: UART B: »
                        <li><b>UART_C</b>: UART C: »
```

html_navigation.html 1,1 Top

"html_navigation.html" 105L, 3178C

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019
#
# monitoring of electronic components
CAPT([hardware])

# A/D (Analog to Digital converter)
ITEM([AD_OFFLINE],              [AD offline])
ITEM([AD_WATCHDOG],             [AD watchdog])

# RTC https://en.wikipedia.org/wiki/Real-time_clock
ITEM([RTC_OFFLINE],             [RTC offline])
ITEM([RTC_NO_SQW],              [no SQW to RTC signa»

# MCU https://en.wikipedia.org/wiki/Microcontroller
ITEM([MCU_WDT],                 [MCU watchdog])

# ETH transceiver
ITEM([ETH_TRANSC_OFFLINE],      [ETH transceiver off»
ITEM([ETH_BUS_ERROR],           [ETH bus integrity e»

# external memory
CAPT([memory])

# DataFlash https://www.adestotech.com/products/data»
ITEM([DFLASH_OFFLINE],          [DataFlash offline])
ITEM([DFLASH_ERASE_WRITE],      [DataFlash erase or »

# FRAM https://en.wikipedia.org/wiki/Ferroelectric_R»
ITEM([FRAM_OFFLINE],            [FRAM offline])
ITEM([FRAM_WRITE],              [FRAM can not write])

# SRAM https://en.wikipedia.org/wiki/Static_random-a»
ITEM([SRAM_OFFLINE],            [SRAM offline])
ITEM([SRAM_WRITE],              [SRAM can not write])

# configuration data integrity error (CRC)
CAPT([conf])

ITEM([AD_CONFIG],               [AD config])
ITEM([MAIN_CONFIG],             [main config])
ITEM([UART_A],                  [UART A: changed add»
ITEM([UART_B],                  [UART B: changed add»
ITEM([UART_C],                  [UART C: changed add»
ITEM([UART_D],                  [UART D: changed add»

# industry protocols settings
CAPT([proto])

ITEM([MBUS_A],                  [M-Bus A: settings c»
ITEM([MBUS_B],                  [M-Bus B: settings c»
ITEM([MBUS_C],                  [M-Bus C: settings c»
ITEM([MBUS_D],                  [M-Bus D: settings c»
ITEM([MODBUS_A],                [MODBUS A: settings »
ITEM([MODBUS_B],                [MODBUS B: settings »
ITEM([MODBUS_C],                [MODBUS C: settings »
ITEM([MODBUS_D],                [MODBUS D: settings »

# tracking user activity
```

installfest_2019.mc 1,1 Top

```
__HEADER([Josef Kubin], [2019/02/17], [embedded_dia»

#                 _____
# --->/ CAPT \---,CAPT
#     \_____/<--'
#                 _____
# --->/ ITEM \---,ITEM
#     \_____/<--'
#
#           ____CAPT____
#          |            |
#  CAPT  _V_  ITEM    __|__
# ----->/ ε \------>/ [,] \---,ITEM
#       \___/       \_____/<--'
# β
define([EPSILON_FIRST], [define([$0], [,])])

# A → β
define([CAPT], [

        # A → β
        define([COMMA], defn([EPSILON_FIRST]))

        divert(0)dnl
undivert(1),
        "$1": LBRAC[]dnl
divert(1)
        RBRAC[]dnl
divert(-1)
])

# A → β
define([ITEM], [

        divert(0)COMMA
                [{"$1": "$2"}]dnl
divert(-1)
])

divert(0)dnl
{"PROJECT": {
        "_comment": "DO_NOT_EDIT"dnl
divert(2)
}}
divert(-1)
~
~
~
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019

CLASS([Bicycle])
#         name      type      getter          setter
PROPERTY([speed],  [int],    [getSpeed],       [setS»
PROPERTY([color],  [String], [getColor])
PROPERTY([gear],   [int],    [getGear],        [setG»
PROPERTY([brand],  [String], [getBrand])

CLASS([Address])
#         name      type      getter          setter
PROPERTY([street], [String], [getStreet])
PROPERTY([city],   [String], [getCity])
PROPERTY([state],  [String], [getState])
PROPERTY([zip],    [String], [getZip])

CLASS([Vegetable])
#         name      type      getter          setter
PROPERTY([name],   [String], [getName])
PROPERTY([kind],   [String], [getKind])
PROPERTY([occurr], [String], [getOccurence],  [setOc»
PROPERTY([color],  [String], [getColor])
PROPERTY([size],   [float],  [getSize],       [setSi»
PROPERTY([edible], [boolean],[getEdible])

CLASS([Animal])
#         name      type      getter          setter
PROPERTY([name],   [String], [getName])
PROPERTY([birth],  [long],   [getBirth])
PROPERTY([wild],   [boolean],[getWildness])
PROPERTY([mamal],  [boolean],[getMamal])
~
~
~
```

M4 and XSLT

- M4
  - Difficult for untrained (ungifted)
  - Straightforward text transformation
    - Direct use: CFG, ATMs, BFRs, stacks
- XSLT
  - Designed for XML transformation
  - Low semantic density (too wordy)
  - No automata

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dependentObject class="Address">
    <property name="street" type="String" getter="ge»
    <property name="city" type="String" getter="getC»
    <property name="state" type="String" getter="get»
    <property name="zip" type="String" getter="getZi»
</dependentObject>
~
~
~
```

Formal Grammar (Chomsky type)

$\mathbf{G = (N, \Sigma, P, S)}$

N: fin. set of non-terminal symbols

Σ: fin. set of terminal symbols

$\mathbf{N \cap \Sigma = \emptyset}$

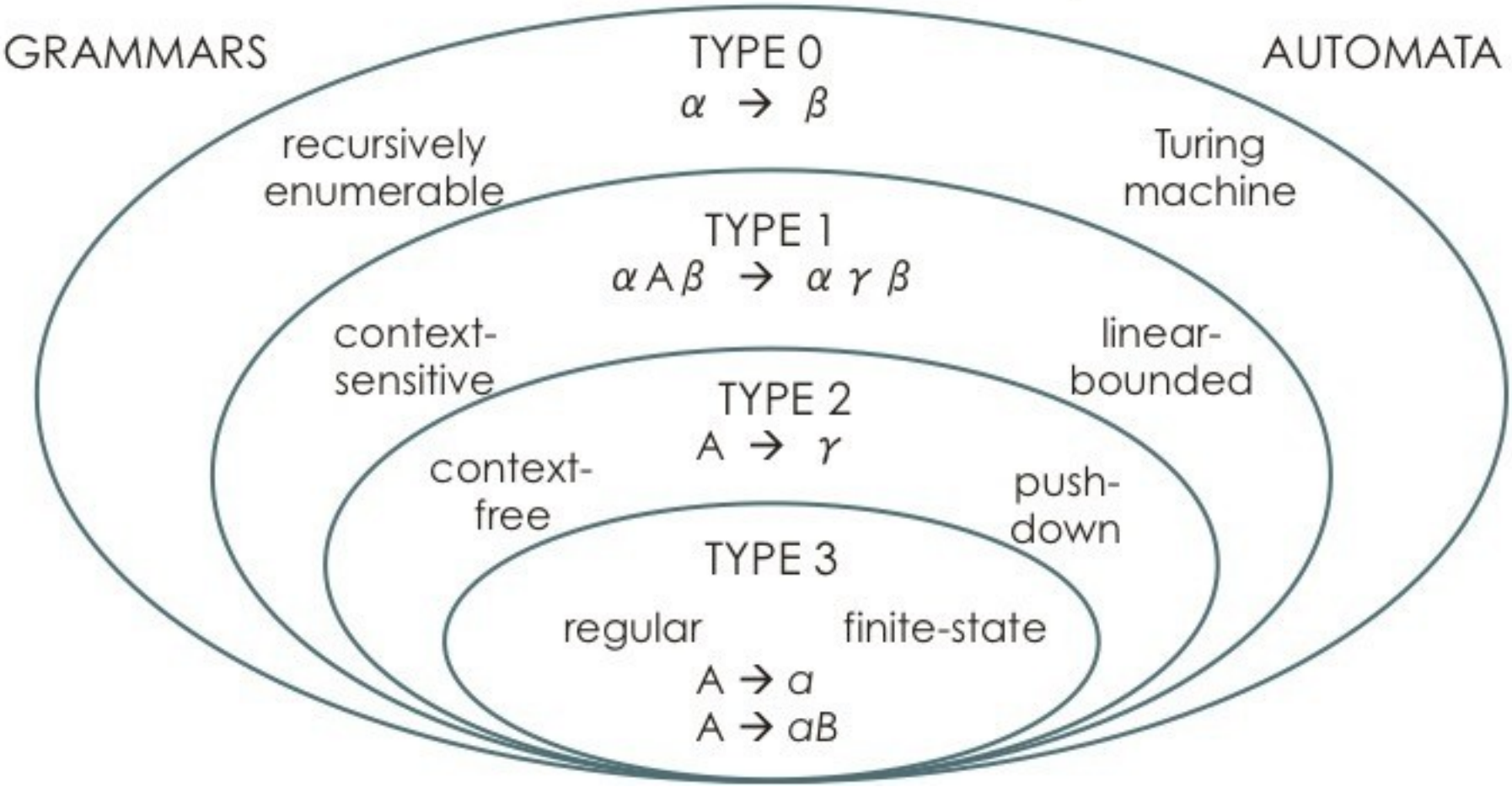
P: fin. set of production (rewrite) rules

$\mathbf{(N \cup \Sigma)^* N (N \cup \Sigma)^* \rightarrow (N \cup \Sigma)^*}$

S: is the start symbol

$\mathbf{S \in N}$

Chomsky Hierarchy



Production rules in M4

- Production rules in Context-Free Grammar (type 2)
  P: A $\rightarrow$ **β**
  A $\in$ **N**
  β $\in$ **(N** $\cup$ **Σ)***

- Productions rules in M4
  define(`**A**', `**β**')

Production rules in M4

Σ = {ε, a, b}; N = {S, A};

A → ε

```
define(`A') or define(`A', ``')
```

A → a

```
define(`A', `a')
```

A → aA

```
define(`A', `a``'A')
```

A → Aa

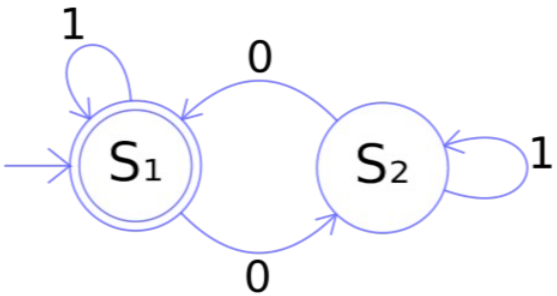
```
define(`A', `A``'a')
```

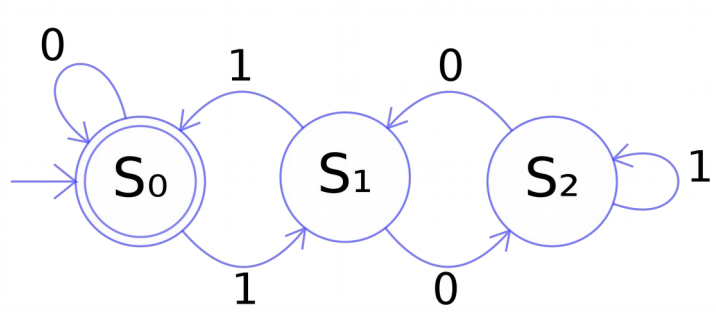
S → aSb

```
define(`S', `a``'S``'b')
```

Automata (abstract machines)

- Pattern matching





- Source code generators
  see my previous Vim screenshots

Buffers (FIFO)

- Buffer size limited by your HDD
- Data in buffers can be dumped or moved between buffers
- The keyword **undivert** without params dumps **ALL** buffers in numeric order: 1, 2, 3, ...
- The keyword **undivert(5, 8, 9)** with param(s) dumps buffers: 5, 8, 9
- The keyword **undivert([file1.txt], [file2.txt], …)** dumps content of files to current buffer stream (NO expansion)

	current buffer stream #ID (settings is global)				
	divert(-1)	divert(0)	divert(1)	divert(2)	divert(n)
undivert(1)	/dev/null	stdout	#1.BFR	#2.BFR	#n.BFR
undivert(2)	/dev/null	stdout	#1.BFR	#2.BFR	#n.BFR
undivert(3)	/dev/null	stdout	#1.BFR	#2.BFR	#n.BFR
...	/dev/null	stdout	#1.BFR	#2.BFR	#n.BFR
undivert dumps all buffers to active stream	/dev/null	stdout	#1.BFR	#2.BFR	#n.BFR

Buffers – a Java example

- Indices for buffers are defined in **buffers.m4**

```
$ pwd
/home/kb/if19/java

$ make
m4 ../master.m4 buffers.m4 java_by_m4.java.m4 config.mc > java_by_m4.java
```

```
# 2019/02/24 Josef Kubin
#
# Source Code Generators
# installfest 2019

CLASS([Bicycle])
#         name      type      getter          setter
PROPERTY([speed],  [int],    [getSpeed],      [setS»
PROPERTY([color],  [String], [getColor])
PROPERTY([gear],   [int],    [getGear],       [setG»
PROPERTY([brand],  [String], [getBrand])

CLASS([Address])
#         name      type      getter          setter
PROPERTY([street], [String], [getStreet])
PROPERTY([city],   [String], [getCity])
PROPERTY([state],  [String], [getState])
PROPERTY([zip],    [String], [getZip])

CLASS([Vegetable])
#         name      type      getter          setter
PROPERTY([name],   [String], [getName])
PROPERTY([kind],   [String], [getKind])
PROPERTY([occurr], [String], [getOccurence],  [setOc»
PROPERTY([color],  [String], [getColor])
PROPERTY([size],   [float],  [getSize],       [setSi»
PROPERTY([edible], [boolean],[getEdible])

CLASS([Animal])
#         name      type      getter          setter
PROPERTY([name],   [String], [getName])
PROPERTY([birth],  [long],   [getBirth])
PROPERTY([wild],   [boolean],[getWildness])
PROPERTY([mamal],  [boolean],[getMamal])
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
config.mc                    1,1           All
```

```
#          _________
# --->/ PROPERTY \---,PROPERTY
#     \__________/<--'
#
#              ______CLASS______
#            |                 |
#  CLASS  _V_  PROPERTY   ___|__
# ------>/ ε \---------->/ [, ] \---,PROPERTY
#        \___/           \______/<--'
# β
define([EPSILON_FIRST], [define([$0], [, ])])

# A → β
define([CLASS], [

        # A → β
        define([COMMA], defn([EPSILON_FIRST]))

        divert(0)dnl
undivert
public class $1 implements java.io.Serializable {
divert(CONSTRUCTOR_NAME)dnl

        public $1(divert(CONSTRUCTOR_ARGS_BRAC)) {
divert(CONSTRUCTOR_BRAC)dnl
        }
divert(FINAL_BRAC)dnl
}
divert(-1)
])

# A → β
define([PROPERTY], [

        divert(PRIVATE_ATTRIB)dnl
        private $2 $1;
divert(CONSTRUCTOR_ARGS)COMMA[]$2 $1[]dnl
divert(CONSTRUCTOR_ASSIGN)dnl
                this.$1 = $1;
divert(GETTERS)
        public $2 $3() {
                return this.$1;
        }
divert(-1)

        # if non-empty setter
        ifelse([$4], [], [], [

                divert(SETTERS)
        public void $4($2 $1) {
                this.$1 = $1;
        }
divert(-1)
        ])
])

divert(0)dnl
/**
 * DO_NOT_EDIT
 */
divert(-1)
java_by_m4.java.m4           63,0-1        Bot
```

```
/**
 * DO NOT EDIT! Generated automatically!
 */

public class Bicycle implements java.io.Serializabl»
        private int speed;
        private String color;
        private int gear;
        private String brand;

        public Bicycle(int speed, String color, int»
                this.speed = speed;
                this.color = color;
                this.gear = gear;
                this.brand = brand;
        }

        public int getSpeed() {
                return this.speed;
        }

        public String getColor() {
                return this.color;
        }

        public int getGear() {
                return this.gear;
        }

        public String getBrand() {
                return this.brand;
        }

        public void setSpeed(int speed) {
                this.speed = speed;
        }

        public void setGear(int gear) {
                this.gear = gear;
        }
}

public class Address implements java.io.Serializabl»
        private String street;
        private String city;
        private String state;
        private String zip;

        public Address(String street, String city, »
                this.street = street;
                this.city = city;
                this.state = state;
                this.zip = zip;
        }

        public String getStreet() {
                return this.street;
        }

        public String getCity() {
                return this.city;
java_by_m4.java              3,3           Top
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

```
__HEADER([Josef Kubin], [2018/10/27], [java])
___DESCR([indices for buffers])

# define counter
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

define([PRIVATE_ATTRIB],                BUFFER_IDX)
define([CONSTRUCTOR_NAME],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS],              BUFFER_IDX)
define([CONSTRUCTOR_ARGS_BRAC],         BUFFER_IDX)
define([CONSTRUCTOR_ASSIGN],            BUFFER_IDX)
define([CONSTRUCTOR_BRAC],              BUFFER_IDX)
define([GETTERS],                       BUFFER_IDX)
define([SETTERS],                       BUFFER_IDX)
define([FINAL_BRAC],                    BUFFER_IDX)
~
~
~
```

Buffers – proverbs example

- Indices for buffers are defined in **proverbs.m4**
- All kind of documents are covered by buffers
- Language specific definitions: **lang_{cz,sk,en,de,it,jp,...}.m4**

```
$ pwd
/home/kb/if19/proverbs

$ make
m4 -DPROJECT=proverbs ../master.m4 proverbs.m4 lang_cz.m4
ini_proverbs.ini.m4 installfest_2019.mc > ini_proverbs.ini
```

```
_HEADER([Josef Kubin], [2018/05/22], [proverbs])
___DESCR([HRUBÝ, Karel. BOOK_NAME Praha: J. OTTO. 18»

# Vim commands for selected paragraph (Visual Inner »
#
# align items to columns
# '<,'>! perl -pe 'use utf8;s/^([^\(]*)/sprintf("\%9»
#
# sort lines
# '<,'>! sort -k 2,2 -t '['
# or
# '<,'>! LC_ALL=C sort -k 2 -t '['

     FARM([A], [Ani kůň netáhne víc než může.])
     FARM([A], [Ani slepice darmo nehrabe.])
   AUGUST([B], [Bylo-li léto mokré, bývá v příštím r»
     FARM([B], [Bitím podaří se jednoho čerta vyhnat»
    MARCH([B], [Březen bez vody, duben bez trávy.])
    MARCH([B], [Březen, za kamna vlezem; duben, ješt»
     FARM([C], [Co doma uvaříš, to doma sněz.])
     FARM([C], [Cvičení a zkoušení dává umění.])
 FEBRUARY([C], [Co do hromnic, to od hromnic, spíše »
 FEBRUARY([C], [Co si únor zazelená, březen si to há»
      MAY([C], [Co červ nesní, to sud polkne (květ v»
 NOVEMBER([C], [Chodí-li husa o sv. Martině po ledě,»
   AUGUST([Č], [Čeho červenec a srpen neuvaří, toho »
     FARM([Č], [Čas růže přináší.])
     FARM([Č], [Častá dohlídka pohání práci.])
     JUNE([Č], [Červen studený, sedlák krčí rameny.])
   SPRING([Č], [Čemu podzim říká: hnilo! tomu jaro p»
    APRIL([D], [Duben hojný vodou, říjen pivem.])
 DECEMBER([D], [Dvanáct nocí od štědrého večera až d»
     FARM([D], [Dá-li Pán Bůh kravičku, dá též travi»
     FARM([D], [Dvojí orání, dvojí chléb, trojí orán»
  JANUARY([D], [Dobrá sanice, koňská smrt.])
     JULY([D], [Den sedmi bratrů když deštivý, bývá »
     JUNE([D], [Dlouhý den, krátká nit.])       <---»
     JUNE([D], [Do sv. Jana Křtitele nechval ječmene»
      MAY([D], [Déší na Mamerta přináší soužení (11.»
      MAY([D], [Déšť v noci před sv. Valburgou naplň»
   SUMMER([D], [Deštivé léto horší podzimku.])
     FARM([H], [Hrne se do toho, jako vůl do kupy se»
     FARM([H], [Hubu zapni a oči napni.])
      MAY([H], [Havlovo žito, Urbanův oves, co z toh»
 NOVEMBER([H], [Hřímá-li v listopadu, bude dobrý rok»
     FARM([I], [I slepá slepice někdy zrnko najde.])
    APRIL([J], [Jak dlouho žába před sv. Jiřím křičí»
    APRIL([J], [Jiří a Marek mrazem nás zalek'.])
 DECEMBER([J], [Jaké počasí na sv. Barboru, takové b»
     FARM([J], [Jaká řepa, taká nať, jaké plášti, ta»
     FARM([J], [Jsou-li v zimě rampouchy hojné, budo»
 FEBRUARY([J], [Je-li v únoru zima a sucho, bývá prý»
  JANUARY([J], [Jaké počasí je který den od vánočníh»
  JANUARY([J], [Jména Ježíš, ke kamnům nejblíž.])
     JUNE([J], [Jaká povětrnost jest na sv. Jana Křt»
     JUNE([J], [Jaké počasí na Ladislava se ukazuje »
    MARCH([J], [Jak jest na popeleční středu, tak bu»
    MARCH([J], [Jedna vlašťovka nedělá léto.])
      MAY([J], [Je-li už máj zahradníkem, není stodo»
 NOVEMBER([J], [Jaký den sv. Kateřiny, takové počasí»
 NOVEMBER([J], [Je-li zamračeno na sv. Martina, bývá»
installfest_2019.mc          1,1          Top
/\<ITEM\>
```

```
__HEADER([Josef Kubin], [2018/10/05], [proverbs])
___POINT([random input, output is sorted by buffer »

#                                        ____JAN,FEB,M»
#                                       |    /
#          _________   JAN,FEB,MAR,...  __V__/
#--->/ CAPTION \---------------->/ ITEM \
#    \_________/                 \______/
#

# β
define([ITEM], [

        divert($0_SCN)dnl
$0_CODE[_]$0_COUNTER="$2"
divert(-1)

        define([$0_COUNTER], incr($0_COUNTER))
])

# β
define([CAPTION], [

        define([$0_COUNTER], [1])

        # transition to the next node
        define([$0], defn([ITEM]))

        divert($0_ABC)
BRAC($0_NAME)
divert(-1)

        # β merge
]defn([ITEM]))

# A → β
define([JANUARY],       defn([CAPTION]))
define([FEBRUARY],      defn([CAPTION]))
define([MARCH],         defn([CAPTION]))
define([APRIL],         defn([CAPTION]))
define([MAY],           defn([CAPTION]))
define([JUNE],          defn([CAPTION]))
define([JULY],          defn([CAPTION]))
define([AUGUST],        defn([CAPTION]))
define([SEPTEMBER],     defn([CAPTION]))
define([OCTOBER],       defn([CAPTION]))
define([NOVEMBER],      defn([CAPTION]))
define([DECEMBER],      defn([CAPTION]))

define([SPRING],        defn([CAPTION]))
define([SUMMER],        defn([CAPTION]))
define([AUTUMN],        defn([CAPTION]))
define([WINTER],        defn([CAPTION]))

define([FARM],          defn([CAPTION]))

divert(0)dnl
; DO_NOT_EDIT
divert(-1)
~
~
ini_proverbs.ini.m4          27,21-28          All
```

```
; DO NOT EDIT! Generated automatically!

[Leden]
jan_1="Dobrá sanice, koňská smrt."
jan_2="Jaké počasí je který den od vánočního Božího»
jan_3="Jména Ježíš, ke kamnům nejblíž."
jan_4="Na nový rok o slepičí krok, na Tři krále o s»
jan_5="Na sv. Pavla na víru obrácení (25.) mlha – p»
jan_6="Pakli leden mokrý bývá, jistě vína nedolívá."
jan_7="Roste den, roste i zima."
jan_8="Slunce svítí na mráz."
jan_9="Velký sníh, malá voda."
jan_10="V lednu-li hrom se ozývá, očekávati lze hoj»
jan_11="Za mlhavým lednem plíží mokré jaro."

[Únor]
feb_1="Co do hromnic, to od hromnic, spíše o lopatu»
feb_2="Co si únor zazelená, březen si to hájí, co s»
feb_3="Je-li v únoru zima a sucho, bývá prý horký s»
feb_4="Kdyby mělo na hromnice mrznout, musí už skři»
feb_5="Krátký masopust, dlouhá zima."
feb_6="Masopust na slunci, pomláska za kamny."
feb_7="Na hromnice je-li vánice, otvírá medvěd boud»
feb_8="Nechce-li severní vítr v únoru váti, v dubnu»
feb_9="O hromnicích musí skřivan vrznout, kdyby měl»
feb_10="O sv. Dorotě (6.) uschne košile na plotě. (»
feb_11="Pakli se bouřkami v únoru rohy býkovi nevik»
feb_12="Přejdou hromnice, konec sanice."
feb_13="Sv. Agata bývá na sníh bohata."
feb_14="Sv. Petr za stolem nechává na holém, a tepr»
feb_15="Teče-li v masopustní úterý voda kolejem, bu»
feb_16="Tmavé hromnice, sedláka radost, jasno-li al»
feb_17="Únor bílý, pole sílí."
feb_18="Větrové severní ku konci února zvěstují úro»

[Březen]
mar_1="Březen bez vody, duben bez trávy."
mar_2="Březen, za kamna vlezem; duben, ještě tam bu»
mar_3="Jak jest na popeleční středu, tak bude celý »
mar_4="Jedna vlašťovka nedělá léto."
mar_5="Kolik březnových mlh, tolik červencových lij»
mar_6="Mnoho dešťů v březnu přivádí hubené léto."
mar_7="Mrzne-li na den čtyřiceti mučedníků, mrzne j»
mar_8="Na Panny Marie zvěstování (25.) kdyby travič»
mar_9="Na sv. Řehoře čáp přeletí přes moře, žába hu»
mar_10="Pěkně-li na sv. Josefa, bývá dobrý rok."
mar_11="Prší-li na suché dny, bude drahota."
mar_12="Strnad volá: Sedláče, sedláče, sít!"
mar_13="Suchý březen, studený duben, mokrý máj, bud»
mar_14="Svítí-li slunce na den zvěstování Panny Mar»
mar_15="Třináctý týden v roce novém zem vyhřeje – p»
mar_16="V březnu prach co zlato a stříbro, ale sníh»

[Duben]
apr_1="Duben hojný vodou, říjen pivem."
apr_2="Jak dlouho žába před sv. Jiřím křičí, rovněž»
apr_3="Jiří a Marek mrazem nás zalek'."
apr_4="Když laštovice letí, jest znamení podletí."
apr_5="Kolik mrazů do sv. Jiří, tolik do sv. Václav»
apr_6="Na bílou sobotu zasazuj štěpy."
apr_7="Na velký pátek zemí nehýbej."
ini_proverbs.ini          1,1          Top
```

```
HEADER([Josef Kubin], [2018/05/22], [proverbs])
DESCR([HRUBÝ, Karel. BOOK_NAME Praha: J. OTTO. 18»

# Vim commands for selected paragraph (Visual Inner »
#
# align items to columns
# '<,'>! perl -pe 'use utf8;s/^([^\(]*)/sprintf("\%9»
#
# sort lines
# '<,'>! sort -k 2,2 -t '['
# or
# '<,'>! LC_ALL=C sort -k 2 -t '['

     FARM([A], [Ani kůň netáhne víc než může.])
     FARM([A], [Ani slepice darmo nehrabe.])
   AUGUST([B], [Bylo-li léto mokré, bývá v příštím r»
     FARM([B], [Bitím podaří se jednoho čerta vyhnat»
    MARCH([B], [Březen bez vody, duben bez trávy.])
    MARCH([B], [Březen, za kamna vlezem; duben, ešť»
     FARM([C], [Co doma uvaříš, to doma sněz.])
     FARM([C], [Cvičení a zkoušení dává umění.])
 FEBRUARY([C], [Co do hromnic, to od hromnic, spíše »
 FEBRUARY([C], [Co si únor zazelená, březen si to há»
      MAY([C], [Co červ nesní, to sud polkne (květ v»
 NOVEMBER([C], [Chodí-li husa o sv. Martině po ledě,»
   AUGUST([Č], [Čeho červenec a srpen neuvaří, toho »
     FARM([Č], [Čas růže přináší.])
     FARM([Č], [Častá dohlídka pohání práci.])
     JUNE([Č], [Červen studený, sedlák krčí rameny.])
   SPRING([Č], [Čemu podzim říká: hnilo! tomu jaro p»
    APRIL([D], [Duben hojný vodou, říjen pivem.])
 DECEMBER([D], [Dvanáct nocí od štědrého večera až d»
     FARM([D], [Dá-li Pán Bůh kravičku, dá též travi»
     FARM([D], [Dvojí orání, dvojí chléb, trojí orán»
  JANUARY([D], [Dobrá sanice, koňská smrt.])
     JULY([D], [Den sedmi bratrů když deštivý, bývá »
     JUNE([D], [Dlouhý den, krátká nit.])       <---»
     JUNE([D], [Do sv. Jana Křtitele nechval ječmene»
      MAY([D], [Déší na Mamerta přináší soužení (11.»
      MAY([D], [Déšť v noci před sv. Valburgou naplň»
   SUMMER([D], [Deštivé léto horší podzimku.])
     FARM([H], [Hrne se do toho, jako vůl do kupy se»
     FARM([H], [Hubu zapni a oči napni.])
      MAY([H], [Havlovo žito, Urbanův oves, co z toh»
 NOVEMBER([H], [Hřímá-li v listopadu, bude dobrý rok»
     FARM([I], [I slepá slepice někdy zrnko najde.])
    APRIL([J], [Jak dlouho žába před sv. Jiřím křičí»
    APRIL([J], [Jiří a Marek mrazem nás zalek'.])
 DECEMBER([J], [Jaké počasí na sv. Barboru, takové b»
     FARM([J], [Jaká řepa, taká nať, jaké plášti, ta»
     FARM([J], [Jsou-li v zimě rampouchy hojné, budo»
 FEBRUARY([J], [Je-li v únoru zima a sucho, bývá prý»
  JANUARY([J], [Jaké počasí je který den od vánočníh»
  JANUARY([J], [Jména Ježíš, ke kamnům nejblíž.])
     JUNE([J], [Jaká povětrnost jest na sv. Jana Křt»
     JUNE([J], [Jaké počasí na Ladislava se ukazuje »
    MARCH([J], [Jak jest na popeleční středu, tak bu»
    MARCH([J], [Jedna vlašťovka nedělá léto.])
      MAY([J], [Je-li už máj zahradníkem, není stodo»
 NOVEMBER([J], [Jaký den sv. Kateřiny, takové počasí»
 NOVEMBER([J], [Je-li zamračeno na sv. Martina, bývá»
installfest_2019.mc          1,2          Top
```

```
HEADER([Josef Kubin], [2018/09/18], [proverbs])
POINT([definition in JANUARY for all])

#                                          ______JA»
#                                         |      /
#          _________   JAN,FEB,MAR,...   _____V____/
# --->/ JANUARY \----------------->/ NEXT_ITEM \
#     \_________/                   \____________/
#

# β
# A → β
define([JANUARY], [

        divert($0_SCN)dnl
<$0_WRAP code="$0_CODE">
        <name>$0_NAME</name>
        <item>$2</item>
divert($0_SCN_END)dnl
</$0_WRAP>
divert(-1)

        # transition to the next node
        define([$0], defn([NEXT_ITEM]))
])

# β
define([NEXT_ITEM], [

        divert($0_SCN)dnl
        <item>$2</item>
divert(-1)
])

define([FEBRUARY],      defn([JANUARY]))
define([MARCH],         defn([JANUARY]))
define([APRIL],         defn([JANUARY]))
define([MAY],           defn([JANUARY]))
define([JUNE],          defn([JANUARY]))
define([JULY],          defn([JANUARY]))
define([AUGUST],        defn([JANUARY]))
define([SEPTEMBER],     defn([JANUARY]))
define([OCTOBER],       defn([JANUARY]))
define([NOVEMBER],      defn([JANUARY]))
define([DECEMBER],      defn([JANUARY]))

define([SPRING],        defn([JANUARY]))
define([SUMMER],        defn([JANUARY]))
define([AUTUMN],        defn([JANUARY]))
define([WINTER],        defn([JANUARY]))

define([FARM],          defn([JANUARY]))

divert(0)dnl
<?xml version="1.0" encoding="utf-8"?>
<PROJECT lang="LANG_CODE">
divert(LAST_BUFFER)dnl
<PROJECT>
divert(-1)
~
~
xml_proverbs.m4              2,2          All
```

```
<?xml version="1.0" encoding="utf-8"?>
<proverbs lang="cz">
<month code="jan">
        <name>Leden</name>
        <item>Dobrá sanice, koňská smrt.</item>
        <item>Jaké počasí je který den od vánočního»
        <item>Jména Ježíš, ke kamnům nejblíž.</item>
        <item>Na nový rok o slepičí krok, na Tři kr»
        <item>Na sv. Pavla na víru obrácení (25.) m»
        <item>Pakli leden mokrý bývá, jistě vína ne»
        <item>Roste den, roste i zima.</item>
        <item>Slunce svítí na mráz.</item>
        <item>Velký sníh, malá voda.</item>
        <item>V lednu-li hrom se ozývá, očekávati l»
        <item>Za mlhavým lednem plíží mokré jaro.</»
</month>
<month code="feb">
        <name>Únor</name>
        <item>Co do hromnic, to od hromnic, spíše o»
        <item>Co si únor zazelená, březen si to háj»
        <item>Je-li v únoru zima a sucho, bývá prý »
        <item>Kdyby mělo na hromnice mrznout, musí »
        <item>Krátký masopust, dlouhá zima.</item>
        <item>Masopust na slunci, pomláska za kamny»
        <item>Na hromnice je-li vánice, otvírá medv»
        <item>Nechce-li severní vítr v únoru váti, »
        <item>O hromnicích musí skřivan vrznout, kd»
        <item>O sv. Dorotě (6.) uschne košile na pl»
        <item>Pakli se bouřkami v únoru rohy býkovi»
        <item>Přejdou hromnice, konec sanice.</item>
        <item>Sv. Agata bývá na sníh bohata.</item>
        <item>Sv. Petr za stolem nechává na holém, »
        <item>Teče-li v masopustní úterý voda kolej»
        <item>Tmavé hromnice, sedláka radost, jasno»
        <item>Únor bílý, pole sílí.</item>
        <item>Větrové severní ku konci února zvěstu»
</month>
<month code="mar">
        <name>Březen</name>
        <item>Březen bez vody, duben bez trávy.</it»
        <item>Březen, za kamna vlezem; duben, ještě»
        <item>Jak jest na popeleční středu, tak bud»
        <item>Jedna vlašťovka nedělá léto.</item>
        <item>Kolik březnových mlh, tolik červencov»
        <item>Mnoho dešťů v březnu přivádí hubené l»
        <item>Mrzne-li na den čtyřiceti mučedníků, »
        <item>Na Panny Marie zvěstování (25.) kdyby»
        <item>Na sv. Řehoře čáp přeletí přes moře, »
        <item>Pěkně-li na sv. Josefa, bývá dobrý ro»
        <item>Prší-li na suché dny, bude drahota.</»
        <item>Strnad volá: Sedláče, sedláče, sít!</»
        <item>Suchý březen, studený duben, mokrý má»
        <item>Svítí-li slunce na den zvěstování Pan»
        <item>Třináctý týden v roce novém zem vyhře»
        <item>V březnu prach co zlato a stříbro, al»
</month>
<month code="apr">
        <name>Duben</name>
        <item>Duben hojný vodou, říjen pivem.</item»
        <item>Jak dlouho žába před sv. Jiřím křičí,»
        <item>Jiří a Marek mrazem nás zalek'.</item»
xml_proverbs_cz.xml          1,1          Top
```

```
_HEADER([Josef Kubin], [2018/05/22], [proverbs])
___DESCR([HRUBÝ, Karel. BOOK_NAME Praha: J. OTTO. 18»

# Vim commands for selected paragraph (Visual Inner »
#
# align items to columns
# '<,'>! perl -pe 'use utf8;s/^([^\(]*)/sprintf("\%9»
#
# sort lines
# '<,'>! sort -k 2,2 -t '['
# or
# '<,'>! LC_ALL=C sort -k 2 -t '['

     FARM([A], [Ani kůň netáhne víc než může.])
     FARM([A], [Ani slepice darmo nehrabe.])
   AUGUST([B], [Bylo-li léto mokré, bývá v příštím r»
     FARM([B], [Bitím podaří se jednoho čerta vyhnat»
    MARCH([B], [Březen bez vody, duben bez trávy.])
    MARCH([B], [Březen, za kamna vlezem; duben, ješt»
     FARM([C], [Co doma uvaříš, to doma sněz.])
     FARM([C], [Cvičení a zkoušení dává umění.])
 FEBRUARY([C], [Co do hromnic, to od hromnic, spíše »
 FEBRUARY([C], [Co si únor zazelená, březen si to há»
      MAY([C], [Co červ nesní, to sud polkne (květ v»
 NOVEMBER([C], [Chodí-li husa o sv. Martině po ledě,»
   AUGUST([Č], [Čeho červenec a srpen neuvaří, toho »
     FARM([Č], [Čas růže přináší.])
     FARM([Č], [Častá dohlídka pohání práci.])
     JUNE([Č], [Červen studený, sedlák krčí rameny.])
   SPRING([Č], [Čemu podzim říká: hnilo! tomu jaro p»
    APRIL([D], [Duben hojný vodou, říjen pivem.])
 DECEMBER([D], [Dvanáct nocí od štědrého večera až d»
     FARM([D], [Dá-li Pán Bůh kravičku, dá též travi»
     FARM([D], [Dvojí orání, dvojí chléb, trojí orán»
  JANUARY([D], [Dobrá sanice, koňská smrt.])
     JULY([D], [Den sedmi bratrů když deštivý, bývá »
     JUNE([D], [Dlouhý den, krátká nit.])       <---»
     JUNE([D], [Do sv. Jana Křtitele nechval ječmene»
      MAY([D], [Déší na Mamerta přináší soužení (11.»
      MAY([D], [Déšť v noci před sv. Valburgou naplň»
   SUMMER([D], [Deštivé léto horší podzimku.])
     FARM([H], [Hrne se do toho, jako vůl do kupy se»
     FARM([H], [Hubu zapni a oči napni.])
      MAY([H], [Havlovo žito, Urbanův oves, co z toh»
 NOVEMBER([H], [Hřímá-li v listopadu, bude dobrý rok»
     FARM([I], [I slepá slepice někdy zrnko najde.])
    APRIL([J], [Jak dlouho žába před sv. Jiřím křičí»
    APRIL([J], [Jiří a Marek mrazem nás zalek'.])
 DECEMBER([J], [Jaké počasí na sv. Barboru, takové b»
     FARM([J], [Jaká řepa, taká nať, jaké plášti, ta»
     FARM([J], [Jsou-li v zimě rampouchy hojné, budo»
 FEBRUARY([J], [Je-li v únoru zima a sucho, bývá prý»
  JANUARY([J], [Jaké počasí je který den od vánočních»
  JANUARY([J], [Jména Ježíš, ke kamnům nejblíž.])
     JUNE([J], [Jaká povětrnost jest na sv. Jana Křt»
     JUNE([J], [Jaké počasí na Ladislava se ukazuje »
    MARCH([J], [Jak jest na popeleční středu, tak bu»
    MARCH([J], [Jedna vlašťovka nedělá léto.])
      MAY([J], [Je-li už máj zahradníkem, není stodo»
 NOVEMBER([J], [Jaký den sv. Kateřiny, takové počasí»
 NOVEMBER([J], [Je-li zamračeno na sv. Martina, bývá»
installfest_2019.mc              1,1           Top
"json_proverbs.json" 269L, 14714C
```

```
__HEADER([Josef Kubin], [2019/02/16], [proverbs])

#                                          _____»
#                                         |     /»
#          _____________   JAN,FEB,MAR,...  _____V____/»
# --->/ FIRST_ITEM \---------------->/ NEXT_ITEM \»
#     \____________/                  \___________/»
#

# β
define([FIRST_ITEM], [

        # transition to the next node
        define([$0], defn([NEXT_ITEM]))

        divert($0_SCN),
        "$0_CODE": LBRAC
                ["$2"]dnl
divert($0_SCN_END)
        RBRAC[]dnl
divert(-1)
])

# β
define([NEXT_ITEM], [

        divert($0_SCN),
                ["$2"]dnl
divert(-1)
])

# A → β
define([JANUARY],       defn([FIRST_ITEM]))
define([FEBRUARY],      defn([FIRST_ITEM]))
define([MARCH],         defn([FIRST_ITEM]))
define([APRIL],         defn([FIRST_ITEM]))
define([MAY],           defn([FIRST_ITEM]))
define([JUNE],          defn([FIRST_ITEM]))
define([JULY],          defn([FIRST_ITEM]))
define([AUGUST],        defn([FIRST_ITEM]))
define([SEPTEMBER],     defn([FIRST_ITEM]))
define([OCTOBER],       defn([FIRST_ITEM]))
define([NOVEMBER],      defn([FIRST_ITEM]))
define([DECEMBER],      defn([FIRST_ITEM]))

define([SPRING],        defn([FIRST_ITEM]))
define([SUMMER],        defn([FIRST_ITEM]))
define([AUTUMN],        defn([FIRST_ITEM]))
define([WINTER],        defn([FIRST_ITEM]))

define([FARM],          defn([FIRST_ITEM]))

divert(0)dnl
{"PROJECT": {
        "_comment": "DO_NOT_EDIT"dnl
divert(LAST_BUFFER)
}}
divert(-1)
~
~
~
json_proverbs.json.m4            15,0-1        All
```

```
{"proverbs": {
        "_comment": "DO NOT EDIT! Generated automat»
        "jan": [
                "Dobrá sanice, koňská smrt.",
                "Jaké počasí je který den od vánočn»
                "Jména Ježíš, ke kamnům nejblíž.",
                "Na nový rok o slepičí krok, na Tři»
                "Na sv. Pavla na víru obrácení (25.»
                "Pakli leden mokrý bývá, jistě vína»
                "Roste den, roste i zima.",
                "Slunce svítí na mráz.",
                "Velký sníh, malá voda.",
                "V lednu-li hrom se ozývá, očekávat»
                "Za mlhavým lednem plíží mokré jaro»
        ],
        "feb": [
                "Co do hromnic, to od hromnic, spíš»
                "Co si únor zazelená, březen si to »
                "Je-li v únoru zima a sucho, bývá p»
                "Kdyby mělo na hromnice mrznout, mu»
                "Krátký masopust, dlouhá zima.",
                "Masopust na slunci, pomláska za ka»
                "Na hromnice je-li vánice, otvírá m»
                "Nechce-li severní vítr v únoru vát»
                "O hromnicích musí skřivan vrznout,»
                "O sv. Dorotě (6.) uschne košile na»
                "Pakli se bouřkami v únoru rohy býk»
                "Přejdou hromnice, konec sanice.",
                "Sv. Agata bývá na sníh bohata.",
                "Sv. Petr za stolem nechává na holé»
                "Teče-li v masopustní úterý voda ko»
                "Tmavé hromnice, sedláka radost, ja»
                "Únor bílý, pole sílí.",
                "Větrové severní ku konci února zvě»
        ],
        "mar": [
                "Březen bez vody, duben bez trávy.",
                "Březen, za kamna vlezem; duben, je»
                "Jak jest na popeleční středu, tak »
                "Jedna vlašťovka nedělá léto.",
                "Kolik březnových mlh, tolik červen»
                "Mnoho dešťů v březnu přivádí huben»
                "Mrzne-li na den čtyřiceti mučedník»
                "Na Panny Marie zvěstování (25.) kd»
                "Na sv. Řehoře čáp přeletí přes moř»
                "Pěkně-li na sv. Josefa, bývá dobrý»
                "Prší-li na suché dny, bude drahota»
                "Strnad volá: Sedláče, sedláče, sít»
                "Suchý březen, studený duben, mokrý»
                "Svítí-li slunce na den zvěstování »
                "Třináctý týden v roce novém zem vy»
                "V březnu prach co zlato a stříbro,»
        ],
        "apr": [
                "Duben hojný vodou, říjen pivem.",
                "Jak dlouho žába před sv. Jiřím kři»
                "Jiří a Marek mrazem nás zalek'.",
                "Když laštovice letí, jest znamení »
                "Kolik mrazů do sv. Jiří, tolik do »
                "Na bílou sobotu zasazuj štěpy.",
                "Na velký pátek zemí nehýbej.",
```

```
json_proverbs.json               1,1           Top
```

```
__HEADER([Josef Kubin], [2018/09/18], [proverbs])

define([AUTHOR], [Karel Hrubý])

# define counter to address buffers
define([BUFFER_IDX], defn([COUNT_UP]))

# init counter
BUFFER_IDX(1)

# indices for description (not used yet)
define([DESCRIPTION_HDR],       BUFFER_IDX)
define([DESCRIPTION_HDR_END],   BUFFER_IDX)

# indices for html, xml, ....
define([START_INDEX_HDR],       BUFFER_IDX)
#
# nothing in between
#
define([JANUARY_HDR],   BUFFER_IDX)
define([FEBRUARY_HDR],  BUFFER_IDX)
define([MARCH_HDR],     BUFFER_IDX)
define([APRIL_HDR],     BUFFER_IDX)
define([MAY_HDR],       BUFFER_IDX)
define([JUNE_HDR],      BUFFER_IDX)
define([JULY_HDR],      BUFFER_IDX)
define([AUGUST_HDR],    BUFFER_IDX)
define([SEPTEMBER_HDR], BUFFER_IDX)
define([OCTOBER_HDR],   BUFFER_IDX)
define([NOVEMBER_HDR],  BUFFER_IDX)
define([DECEMBER_HDR],  BUFFER_IDX)

define([AFTER_MONTHS],  BUFFER_IDX)

define([SPRING_HDR],    BUFFER_IDX)
define([SUMMER_HDR],    BUFFER_IDX)
define([AUTUMN_HDR],    BUFFER_IDX)
define([WINTER_HDR],    BUFFER_IDX)

define([AFTER_SEASONS], BUFFER_IDX)

define([FARM_HDR],      BUFFER_IDX)

define([DECLARATIONS],  BUFFER_IDX)

define([CLOSE_NAVIG],   BUFFER_IDX)

define([JANUARY_ABC],   BUFFER_IDX)
define([JANUARY_ABC_END],       BUFFER_IDX)
define([JANUARY_SCN],   BUFFER_IDX)
define([JANUARY_SCN_END],       BUFFER_IDX)
define([FEBRUARY_ABC],  BUFFER_IDX)
define([FEBRUARY_ABC_END],      BUFFER_IDX)
define([FEBRUARY_SCN],  BUFFER_IDX)
define([FEBRUARY_SCN_END],      BUFFER_IDX)
define([MARCH_ABC],     BUFFER_IDX)
define([MARCH_ABC_END], BUFFER_IDX)
define([MARCH_SCN],     BUFFER_IDX)
define([MARCH_SCN_END], BUFFER_IDX)
define([APRIL_ABC],     BUFFER_IDX)
define([APRIL_ABC_END], BUFFER_IDX)
```

proverbs.m4 5,1 Top

```
define([JULY_SCN],      BUFFER_IDX)
define([JULY_SCN_END],  BUFFER_IDX)
define([AUGUST_ABC],    BUFFER_IDX)
define([AUGUST_ABC_END],        BUFFER_IDX)
define([AUGUST_SCN],    BUFFER_IDX)
define([AUGUST_SCN_END],        BUFFER_IDX)
define([SEPTEMBER_ABC], BUFFER_IDX)
define([SEPTEMBER_ABC_END],     BUFFER_IDX)
define([SEPTEMBER_SCN], BUFFER_IDX)
define([SEPTEMBER_SCN_END],     BUFFER_IDX)
define([OCTOBER_ABC],   BUFFER_IDX)
define([OCTOBER_ABC_END],       BUFFER_IDX)
define([OCTOBER_SCN],   BUFFER_IDX)
define([OCTOBER_SCN_END],       BUFFER_IDX)
define([NOVEMBER_ABC],  BUFFER_IDX)
define([NOVEMBER_ABC_END],      BUFFER_IDX)
define([NOVEMBER_SCN],  BUFFER_IDX)
define([NOVEMBER_SCN_END],      BUFFER_IDX)
define([DECEMBER_ABC],  BUFFER_IDX)
define([DECEMBER_ABC_END],      BUFFER_IDX)
define([DECEMBER_SCN],  BUFFER_IDX)
define([DECEMBER_SCN_END],      BUFFER_IDX)

define([SPRING_ABC],    BUFFER_IDX)
define([SPRING_ABC_END],        BUFFER_IDX)
define([SPRING_SCN],    BUFFER_IDX)
define([SPRING_SCN_END],        BUFFER_IDX)
define([SUMMER_ABC],    BUFFER_IDX)
define([SUMMER_ABC_END],        BUFFER_IDX)
define([SUMMER_SCN],    BUFFER_IDX)
define([SUMMER_SCN_END],        BUFFER_IDX)
define([AUTUMN_ABC],    BUFFER_IDX)
define([AUTUMN_ABC_END],        BUFFER_IDX)
define([AUTUMN_SCN],    BUFFER_IDX)
define([AUTUMN_SCN_END],        BUFFER_IDX)
define([WINTER_ABC],    BUFFER_IDX)
define([WINTER_ABC_END],        BUFFER_IDX)
define([WINTER_SCN],    BUFFER_IDX)
define([WINTER_SCN_END],        BUFFER_IDX)

define([FARM_ABC],      BUFFER_IDX)
define([FARM_ABC_END],  BUFFER_IDX)
define([FARM_SCN],      BUFFER_IDX)
define([FARM_SCN_END],  BUFFER_IDX)

# indices for array of pointers
define([START_INDEX_PTR],       BUFFER_IDX)
#
# nothing in between
#
define([JANUARY_PTR],   BUFFER_IDX)
define([FEBRUARY_PTR],  BUFFER_IDX)
define([MARCH_PTR],     BUFFER_IDX)
define([APRIL_PTR],     BUFFER_IDX)
define([MAY_PTR],       BUFFER_IDX)
define([JUNE_PTR],      BUFFER_IDX)
define([JULY_PTR],      BUFFER_IDX)
define([AUGUST_PTR],    BUFFER_IDX)
define([SEPTEMBER_PTR], BUFFER_IDX)
define([OCTOBER_PTR],   BUFFER_IDX)
define([NOVEMBER_PTR],  BUFFER_IDX)
```

proverbs.m4 74,9 55%

```
define([NOVEMBER_PTR],  BUFFER_IDX)
define([DECEMBER_PTR],  BUFFER_IDX)

define([SPRING_PTR],    BUFFER_IDX)
define([SUMMER_PTR],    BUFFER_IDX)
define([AUTUMN_PTR],    BUFFER_IDX)
define([WINTER_PTR],    BUFFER_IDX)

define([FARM_PTR],      BUFFER_IDX)

define([LAST_BUFFER],   BUFFER_IDX)

# forget unused macros (if you want to be perfect .»
undefine(

        [BUFFER_IDX],

)

# define short codes for items
define([JANUARY_CODE],          [jan])
define([FEBRUARY_CODE],         [feb])
define([MARCH_CODE],            [mar])
define([APRIL_CODE],            [apr])
define([MAY_CODE],              [may])
define([JUNE_CODE],             [jun])
define([JULY_CODE],             [jul])
define([AUGUST_CODE],           [aug])
define([SEPTEMBER_CODE],        [sep])
define([OCTOBER_CODE],          [oct])
define([NOVEMBER_CODE],         [nov])
define([DECEMBER_CODE],         [dec])

define([SPRING_CODE],           [spr])
define([SUMMER_CODE],           [sum])
define([AUTUMN_CODE],           [aut])
define([WINTER_CODE],           [wint])

define([FARM_CODE],             [farm])

# xml wrapping tags
define([JANUARY_WRAP],          [month])
define([FEBRUARY_WRAP],         defn([JANUARY_WRAP]»
define([MARCH_WRAP],            defn([JANUARY_WRAP]»
define([APRIL_WRAP],            defn([JANUARY_WRAP]»
define([MAY_WRAP],              defn([JANUARY_WRAP]»
define([JUNE_WRAP],             defn([JANUARY_WRAP]»
define([JULY_WRAP],             defn([JANUARY_WRAP]»
define([AUGUST_WRAP],           defn([JANUARY_WRAP]»
define([SEPTEMBER_WRAP],        defn([JANUARY_WRAP]»
define([OCTOBER_WRAP],          defn([JANUARY_WRAP]»
define([NOVEMBER_WRAP],         defn([JANUARY_WRAP]»
define([DECEMBER_WRAP],         defn([JANUARY_WRAP]»

define([SPRING_WRAP],           [season])
define([SUMMER_WRAP],           defn([SPRING_WRAP]))
define([AUTUMN_WRAP],           defn([SPRING_WRAP]))
define([WINTER_WRAP],           defn([SPRING_WRAP]))

define([FARM_WRAP],             [farm])
~
```

proverbs.m4 193,1 Bot

```
define([PROJECT_NAME],          [Generátory zdrojové»
define([LANG_cz],               [Čeština])
define([LANG_sk],               [Slovenština])
define([LANG_en],               [Angličtina])
define([LANG_pl],               [Polština])
define([LANG_fi],               [Finština])
define([LANG_hu],               [Maďarština])
define([LANG_de],               [Němčina])
define([LANG_ru],               [Ruština])
define([LANG_uk],               [Ukrajinština])
define([LANG_fr],               [Francouzština])
define([LANG_it],               [Italština])
define([LANG_hi],               [Hindština])
define([LANG_mn],               [Mongolština])
define([LANG_ja],               [Japonština])
define([LANG_zh_CN],            [Čínština (zjednoduš»
define([LANG_so],               [Somálština])
define([LANG_sw],               [Svahilština])
define([LANG_yi],               [Jidiš])
define([LANG_fa],               [Perština])
define([MONTHS],                [Měsíce])
define([SEASONS],               [Období])
define([THE_OTHERS],            [Ostatní])
define([JANUARY_NAME],          [Leden])
define([FEBRUARY_NAME],         [Únor])
define([MARCH_NAME],            [Březen])
define([APRIL_NAME],            [Duben])
define([MAY_NAME],              [Květen])
define([JUNE_NAME],             [Červen])
define([JULY_NAME],             [Červenec])
define([AUGUST_NAME],           [Srpen])
define([SEPTEMBER_NAME],        [Září])
define([OCTOBER_NAME],          [Říjen])
define([NOVEMBER_NAME],         [Listopad])
define([DECEMBER_NAME],         [Prosinec])
define([SPRING_NAME],           [Jaro])
define([SUMMER_NAME],           [Léto])
define([AUTUMN_NAME],           [Podzim])
define([WINTER_NAME],           [Zima])
define([FARM_NAME],             [Hospodářská přísloví»
define([BOOK_NAME],             [Sbírka přísloví, po»
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
lang_cz.m4                    1,1           All
```

```
define([PROJECT_NAME],          [Source code genera»
define([LANG_cz],               [Czech])
define([LANG_sk],               [Slovak])
define([LANG_en],               [English])
define([LANG_pl],               [Polish])
define([LANG_fi],               [Finnish])
define([LANG_hu],               [Hungarian])
define([LANG_de],               [German])
define([LANG_ru],               [Russian])
define([LANG_uk],               [Ukrainian])
define([LANG_fr],               [French])
define([LANG_it],               [Italian])
define([LANG_hi],               [Hindi])
define([LANG_mn],               [Mongolian])
define([LANG_ja],               [Japanese])
define([LANG_zh_CN],            [Chinese (Simplifie»
define([LANG_so],               [Somalia])
define([LANG_sw],               [Swahili])
define([LANG_yi],               [Yiddish])
define([LANG_fa],               [Persian])
define([MONTHS],                [Months])
define([SEASONS],               [Period])
define([THE_OTHERS],            [Other])
define([JANUARY_NAME],          [January])
define([FEBRUARY_NAME],         [February])
define([MARCH_NAME],            [March])
define([APRIL_NAME],            [April])
define([MAY_NAME],              [May])
define([JUNE_NAME],             [June])
define([JULY_NAME],             [July])
define([AUGUST_NAME],           [August])
define([SEPTEMBER_NAME],        [September])
define([OCTOBER_NAME],          [October])
define([NOVEMBER_NAME],         [November])
define([DECEMBER_NAME],         [December])
define([SPRING_NAME],           [The Spring])
define([SUMMER_NAME],           [Summer])
define([AUTUMN_NAME],           [Autumn])
define([WINTER_NAME],           [Winter])
define([FARM_NAME],             [Farm proverb])
define([BOOK_NAME],             [A collection of pr»
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
lang_en.m4                    1,1           All
```

```
define([PROJECT_NAME],          [ソースコードジェネ»
define([LANG_cz],               [チェコ語])
define([LANG_sk],               [スロバキア])
define([LANG_en],               [英語])
define([LANG_pl],               [研磨])
define([LANG_fi],               [フィンランド語])
define([LANG_hu],               [ハンガリー語])
define([LANG_de],               [ドイツ人])
define([LANG_ru],               [ロシア])
define([LANG_uk],               [ウクライナ語])
define([LANG_fr],               [フランス語])
define([LANG_it],               [イタリアの])
define([LANG_hi],               [ヒンディー語])
define([LANG_mn],               [モンゴル語])
define([LANG_ja],               [日本人])
define([LANG_zh_CN],            [中国語（簡体字）])
define([LANG_so],               [ソマリア])
define([LANG_sw],               [スワヒリ語])
define([LANG_yi],               [イディッシュ語])
define([LANG_fa],               [ペルシャ語])
define([MONTHS],                [月])
define([SEASONS],               [期間])
define([THE_OTHERS],            [その他の])
define([JANUARY_NAME],          [1月])
define([FEBRUARY_NAME],         [2月])
define([MARCH_NAME],            [行進])
define([APRIL_NAME],            [4月])
define([MAY_NAME],              [5月])
define([JUNE_NAME],             [六月])
define([JULY_NAME],             [7月])
define([AUGUST_NAME],           [8月])
define([SEPTEMBER_NAME],        [9月])
define([OCTOBER_NAME],          [10月])
define([NOVEMBER_NAME],         [11月])
define([DECEMBER_NAME],         [12月])
define([SPRING_NAME],           [春])
define([SUMMER_NAME],           [夏])
define([AUTUMN_NAME],           [秋])
define([WINTER_NAME],           [冬])
define([FARM_NAME],             [農場のことわざ])
define([BOOK_NAME],             [私たちの農民が使う»
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
lang_ja.m4                    1,1           All
"lang_ja.m4" 41L, 1527C
```

```
_HEADER([Josef Kubin], [2018/05/22], [proverbs])
___DESCR([HRUBÝ, Karel. BOOK_NAME Praha: J. OTTO. 18»

# Vim commands for selected paragraph (Visual Inner »
#
# align items to columns
# '<,'>! perl -pe 'use utf8;s/^([^\(]*)/sprintf("\%9»
#
# sort lines
# '<,'>! sort -k 2,2 -t '['
# or
# '<,'>! LC_ALL=C sort -k 2 -t '['

     FARM([A], [Ani kůň netáhne víc než může.])
     FARM([A], [Ani slepice darmo nehrabe.])
   AUGUST([B], [Bylo-li léto mokré, bývá v příštím r»
     FARM([B], [Bitím podaří se jednoho čerta vyhnat»
    MARCH([B], [Březen bez vody, duben bez trávy.])
    MARCH([B], [Březen, za kamna vlezem; duben, ješt»
     FARM([C], [Co doma uvaříš, to doma sněz.])
     FARM([C], [Cvičení a zkoušení dává umění.])
 FEBRUARY([C], [Co do hromnic, to od hromnic, spíše »
 FEBRUARY([C], [Co si únor zazelená, březen si to há»
      MAY([C], [Co červ nesní, to sud polkne (květ v»
 NOVEMBER([C], [Chodí-li husa o sv. Martině po ledě,»
   AUGUST([Č], [Čeho červenec a srpen neuvaří, toho »
     FARM([Č], [Čas růže přináší.])
     FARM([Č], [Častá dohlídka pohání práci.])
     JUNE([Č], [Červen studený, sedlák krčí rameny.])
   SPRING([Č], [Čemu podzim říká: hnilo! tomu jaro p»
    APRIL([D], [Duben hojný vodou, říjen pivem.])
 DECEMBER([D], [Dvanáct nocí od štědrého večera až d»
     FARM([D], [Dá-li Pán Bůh kravičku, dá též travi»
     FARM([D], [Dvojí orání, dvojí chléb, trojí orán»
  JANUARY([D], [Dobrá sanice, koňská smrt.])
     JULY([D], [Den sedmi bratrů když deštivý, bývá »
     JUNE([D], [Dlouhý den, krátká nit.])       <---»
     JUNE([D], [Do sv. Jana Křtitele nechval ječmene»
      MAY([D], [Déší na Mamerta přináší soužení (11.»
      MAY([D], [Déšť v noci před sv. Valburgou naplň»
   SUMMER([D], [Deštivé léto horší podzimku.])
     FARM([H], [Hrne se do toho, jako vůl do kupy se»
     FARM([H], [Hubu zapni a oči napni.])
      MAY([H], [Havlovo žito, Urbanův oves, co z toh»
 NOVEMBER([H], [Hřímá-li v listopadu, bude dobrý rok»
     FARM([I], [I slepá slepice někdy zrnko najde.])
    APRIL([J], [Jak dlouho žába před sv. Jiřím křičí»
    APRIL([J], [Jiří a Marek mrazem nás zalek'.])
 DECEMBER([J], [Jaké počasí na sv. Barboru, takové b»
     FARM([J], [Jaká řepa, taká nať, jaké plášti, ta»
     FARM([J], [Jsou-li v zimě rampouchy hojné, budo»
 FEBRUARY([J], [Je-li v únoru zima a sucho, bývá prý»
  JANUARY([J], [Jaké počasí je který den od vánočníh»
  JANUARY([J], [Jména Ježíš, ke kamnům nejblíž.])
     JUNE([J], [Jaká povětrnost jest na sv. Jana Křt»
     JUNE([J], [Jaké počasí na Ladislava se ukazuje »
    MARCH([J], [Jak jest na popeleční středu, tak bu»
    MARCH([J], [Jedna vlašťovka nedělá léto.])
      MAY([J], [Je-li už máj zahradníkem, není stodo»
 NOVEMBER([J], [Jaký den sv. Kateřiny, takové počasí»
 NOVEMBER([J], [Je-li zamračeno na sv. Martina, bývá»
installfest_2019.mc                 1,1           Top
```

```
# vim: set ft=m4:
# ^^^^ this is Vim modeline (to force M4 filetype)
__HEADER([Josef Kubin], [2018/10/04], [proverbs])

#                                                 _____»
#                                                |     /
#          _____________   JAN,FEB,MAR,...    _____V____/
# --->/ FIRST_ITEM \----------------->/ NEXT_ITEM \
#     \____________/                  \___________/
#

# β
define([FIRST_ITEM], [

        define([ANCHOR], $0_CODE[]_$1)

        divert($0_HDR)dnl
                <li><a href="[#]$0_CODE">$0_NAME</a»
divert($0_ABC)dnl
        <section>
                <h2 id="$0_CODE">$0_NAME</h2>   <!-»
                <ul>
                        <li><a href="[#]ANCHOR">$1<»
divert($0_ABC_END)dnl
                </ul>
divert($0_SCN)dnl
                <ol>
                        <li id="ANCHOR">$2</li> <!-»
divert($0_SCN_END)dnl
                </ol>
        </section>
divert(-1)

        # transition to the next node
        define([$0], defn([NEXT_ITEM]))
])

# β
define([NEXT_ITEM], [

        ifelse(ANCHOR, $0_CODE[]_$1, [
                divert($0_SCN)dnl
                        <li>$2</li>
divert(-1)
        ], [
                define([ANCHOR], $0_CODE[]_$1)

                divert($0_ABC)dnl
                        <li><a href="[#]ANCHOR">$1<»
divert($0_SCN)dnl
                        <li id="ANCHOR">$2</li>
divert(-1)
        ])
])

# A → β
define([JANUARY],       defn([FIRST_ITEM]))
define([FEBRUARY],      defn([FIRST_ITEM]))
define([MARCH],         defn([FIRST_ITEM]))
define([APRIL],         defn([FIRST_ITEM]))
define([MAY],           defn([FIRST_ITEM]))
html_proverbs.m4                    1,1           Top
```

```
<!DOCTYPE html>
<!-- DO NOT EDIT! Generated automatically! -->
<html>
<head>
<meta charset="UTF-8">
<title>Karel Hrubý</title>
<style>
html {
        font-size: 120%;
}
section ul li {
        display: inline;
}
</style>
</head>
<body>
<header>
<b style="color:red">Čeština</b> (cz) |
<a href="html_proverbs_sk.html">Slovenština</a> (sk»
<a href="html_proverbs_en.html">Angličtina</a> (en)»
<a href="html_proverbs_pl.html">Polština</a> (pl) |
<a href="html_proverbs_fi.html">Finština</a> (fi) |
<a href="html_proverbs_hu.html">Maďarština</a> (hu)»
<a href="html_proverbs_de.html">Němčina</a> (de) |
<a href="html_proverbs_ru.html">Ruština</a> (ru) |
<a href="html_proverbs_uk.html">Ukrajinština</a> (u»
<a href="html_proverbs_fr.html">Francouzština</a> (»
<a href="html_proverbs_it.html">Italština</a> (it) |
<a href="html_proverbs_hi.html">Hindština</a> (hi) |
<a href="html_proverbs_mn.html">Mongolština</a> (mn»
<a href="html_proverbs_ja.html">Japonština</a> (ja)»
<a href="html_proverbs_zh_CN.html">Čínština (zjedno»
<a href="html_proverbs_so.html">Somálština</a> (so)»
<a href="html_proverbs_sw.html">Svahilština</a> (sw»
<a href="html_proverbs_yi.html">Jidiš</a> (yi) |
<a href="html_proverbs_fa.html">Perština</a> (fa) |
<h1>Karel Hrubý</h1>
<p>HRUBÝ, Karel. <b>Sbírka přísloví, pořekadel a pr»
</header>
<nav>
        <h2>Měsíce</h2>
        <ul>
                <li><a href="#jan">Leden</a></li>  »
                <li><a href="#feb">Únor</a></li>   »
                <li><a href="#mar">Březen</a></li» »
                <li><a href="#apr">Duben</a></li>  »
                <li><a href="#may">Květen</a></li» »
                <li><a href="#jun">Červen</a></li» »
                <li><a href="#jul">Červenec</a></li»
                <li><a href="#aug">Srpen</a></li>  »
                <li><a href="#sep">Září</a></li>   »
                <li><a href="#oct">Říjen</a></li>  »
                <li><a href="#nov">Listopad</a></li»
                <li><a href="#dec">Prosinec</a></li»
        </ul>
        <h2>Období</h2>
        <ul>
                <li><a href="#spr">Jaro</a></li>   »
                <li><a href="#sum">Léto</a></li>   »
                <li><a href="#aut">Podzim</a></li» »
                <li><a href="#wint">Zima</a></li>  »
html_proverbs_cz.html               1,1           Top
```

"html_proverbs_cz.html" 572L, 27240C

```
_HEADER([Josef Kubin], [2018/05/22], [proverbs])
__DESCR([HRUBÝ, Karel. BOOK_NAME Praha: J. OTTO. 18»

# Vim commands for selected paragraph (Visual Inner »
#
# align items to columns
# '<,'>! perl -pe 'use utf8;s/^([^\(]*)/sprintf("\%9»
#
# sort lines
# '<,'>! sort -k 2,2 -t '['
# or
# '<,'>! LC_ALL=C sort -k 2 -t '['

     FARM([A], [Ani kůň netáhne víc než může.])
     FARM([A], [Ani slepice darmo nehrabe.])
   AUGUST([B], [Bylo-li léto mokré, bývá v příštím r»
     FARM([B], [Bitím podaří se jednoho čerta vyhnat»
    MARCH([B], [Březen bez vody, duben bez trávy.])
    MARCH([B], [Březen, za kamna vlezem; duben, ješt»
     FARM([C], [Co doma uvaříš, to doma sněz.])
     FARM([C], [Cvičení a zkoušení dává umění.])
 FEBRUARY([C], [Co do hromnic, to od hromnic, spíše »
 FEBRUARY([C], [Co si únor zazelená, březen si to há»
      MAY([C], [Co červ nesní, to sud polkne (květ v»
 NOVEMBER([C], [Chodí-li husa o sv. Martině po ledě,»
   AUGUST([Č], [Čeho červenec a srpen neuvaří, toho »
     FARM([Č], [Čas růže přináší.])
     FARM([Č], [Častá dohlídka pohání práci.])
     JUNE([Č], [Červen studený, sedlák krčí rameny.])
   SPRING([Č], [Čemu podzim říká: hnilo! tomu jaro p»
    APRIL([D], [Duben hojný vodou, říjen pivem.])
 DECEMBER([D], [Dvanáct nocí od štědrého večera až d»
     FARM([D], [Dá-li Pán Bůh kravičku, dá též travi»
     FARM([D], [Dvojí orání, dvojí chléb, trojí orán»
  JANUARY([D], [Dobrá sanice, koňská smrt.])
     JULY([D], [Den sedmi bratrů když deštivý, bývá »
     JUNE([D], [Dlouhý den, krátká nit.])       <---»
     JUNE([D], [Do sv. Jana Křtitele nechval ječmene»
      MAY([D], [Déší na Mamerta přináší soužení (11.»
      MAY([D], [Déšť v noci před sv. Valburgou naplň»
   SUMMER([D], [Deštivé léto horší podzimku.])
     FARM([H], [Hrne se do toho, jako vůl do kupy se»
     FARM([H], [Hubu zapni a oči napni.])
      MAY([H], [Havlovo žito, Urbanův oves, co z toh»
 NOVEMBER([H], [Hřímá-li v listopadu, bude dobrý rok»
     FARM([I], [I slepá slepice někdy zrnko najde.])
    APRIL([J], [Jak dlouho žába před sv. Jiřím křičí»
    APRIL([J], [Jiří a Marek mrazem nás zalek'.])
 DECEMBER([J], [Jaké počasí na sv. Barboru, takové b»
     FARM([J], [Jaká řepa, taká nať, jaké plášti, ta»
     FARM([J], [Jsou-li v zimě rampouchy hojné, budo»
 FEBRUARY([J], [Je-li v únoru zima a sucho, bývá prý»
  JANUARY([J], [Jaké počasí je který den od vánočníh»
  JANUARY([J], [Jména Ježíš, ke kamnům nejblíž.])
     JUNE([J], [Jaká povětrnost jest na sv. Jana Křt»
     JUNE([J], [Jaké počasí na Ladislava se ukazuje »
    MARCH([J], [Jak jest na popeleční středu, tak bu»
    MARCH([J], [Jedna vlašťovka nedělá léto.])
      MAY([J], [Je-li už máj zahradníkem, není stodo»
 NOVEMBER([J], [Jaký den sv. Kateřiny, takové počasí»
 NOVEMBER([J], [Je-li zamračeno na sv. Martina, bývá»
installfest_2019.mc                 1,1          Top
```

```
# vim: set ft=m4:
# ^^^^ this is Vim modeline (to force M4 filetype)
__HEADER([Josef Kubin], [2018/10/04], [proverbs])

#                                                 _____»
#                                                |     /
#          _____________   JAN,FEB,MAR,...    _____V____/
# --->/ FIRST_ITEM \----------------->/ NEXT_ITEM \
#     \____________/                  \___________/
#

# β
define([FIRST_ITEM], [

        define([ANCHOR], $0_CODE[]_$1)

        divert($0_HDR)dnl
                <li><a href="[#]$0_CODE">$0_NAME</a»
divert($0_ABC)dnl
        <section>
                <h2 id="$0_CODE">$0_NAME</h2>   <!-»
                <ul>
                        <li><a href="[#]ANCHOR">$1<»
divert($0_ABC_END)dnl
                </ul>
divert($0_SCN)dnl
                <ol>
                        <li id="ANCHOR">$2</li> <!-»
divert($0_SCN_END)dnl
                </ol>
        </section>
divert(-1)

        # transition to the next node
        define([$0], defn([NEXT_ITEM]))
])

# β
define([NEXT_ITEM], [

        ifelse(ANCHOR, $0_CODE[]_$1, [
                divert($0_SCN)dnl
                        <li>$2</li>
divert(-1)
        ], [
                define([ANCHOR], $0_CODE[]_$1)

                divert($0_ABC)dnl
                        <li><a href="[#]ANCHOR">$1<»
divert($0_SCN)dnl
                        <li id="ANCHOR">$2</li>
divert(-1)
        ])
])

# A → β
define([JANUARY],       defn([FIRST_ITEM]))
define([FEBRUARY],      defn([FIRST_ITEM]))
define([MARCH],         defn([FIRST_ITEM]))
define([APRIL],         defn([FIRST_ITEM]))
define([MAY],           defn([FIRST_ITEM]))
html_proverbs.m4                    2,1          Top
```

```
<!DOCTYPE html>
<!-- DO NOT EDIT! Generated automatically! -->
<html>
<head>
<meta charset="UTF-8">
<title>Karel Hrubý</title>
<style>
html {
        font-size: 120%;
}
section ul li {
        display: inline;
}
</style>
</head>
<body>
<header>
<a href="html_proverbs_cz.html">Český</a> (cz) |
<b style="color:red">Slovenský</b> (sk) |
<a href="html_proverbs_en.html">Angličtina</a> (en)»
<a href="html_proverbs_pl.html">Poľský</a> (pl) |
<a href="html_proverbs_fi.html">Fínsky</a> (fi) |
<a href="html_proverbs_hu.html">Maďarský</a> (hu) |
<a href="html_proverbs_de.html">Nemec</a> (de) |
<a href="html_proverbs_ru.html">Ruský</a> (ru) |
<a href="html_proverbs_uk.html">Ukrajinský</a> (uk)»
<a href="html_proverbs_fr.html">Francúzsky</a> (fr)»
<a href="html_proverbs_it.html">Taliansky</a> (it) |
<a href="html_proverbs_hi.html">Hindčina</a> (hi) |
<a href="html_proverbs_mn.html">Mongolský</a> (mn) |
<a href="html_proverbs_ja.html">Japonský</a> (ja) |
<a href="html_proverbs_zh_CN.html">Zjednodušená čín»
<a href="html_proverbs_so.html">Somálsko</a> (so) |
<a href="html_proverbs_sw.html">Swahilčina</a> (sw)»
<a href="html_proverbs_yi.html">Jidiš</a> (yi) |
<a href="html_proverbs_fa.html">Perzský</a> (fa) |
<h1>Karel Hrubý</h1>
<p>HRUBÝ, Karel. <b>Zbierka príslovie, porekadiel a»
</header>
<nav>
        <h2>Mesiaca</h2>
        <ul>
                <li><a href="#jan">Január</a></li> »
                <li><a href="#feb">Február</a></li»
                <li><a href="#mar">Marec</a></li>  »
                <li><a href="#apr">Apríl</a></li>  »
                <li><a href="#may">Smieť</a></li>  »
                <li><a href="#jun">Jún</a></li> <!-»
                <li><a href="#jul">Júl</a></li> <!-»
                <li><a href="#aug">August</a></li» »
                <li><a href="#sep">Septembra</a></l»
                <li><a href="#oct">Október</a></li»
                <li><a href="#nov">November</a></li»
                <li><a href="#dec">December</a></li»
        </ul>
        <h2>Perióda</h2>
        <ul>
                <li><a href="#spr">Pružina</a></li»
                <li><a href="#sum">Letné</a></li>  »
                <li><a href="#aut">Jeseň</a></li>  »
                <li><a href="#wint">Zimné</a></li» »
html_proverbs_sk.html               12,17-24     Top
"html_proverbs_sk.html" 572L, 27224C
```

```
__HEADER([Josef Kubin], [2018/05/22], [proverbs])
___DESCR([HRUBÝ, Karel. BOOK_NAME Praha: J. OTTO. 18»

# Vim commands for selected paragraph (Visual Inner »
#
# align items to columns
# '<,'>! perl -pe 'use utf8;s/^([^\(]*)/sprintf("\%9»
#
# sort lines
# '<,'>! sort -k 2,2 -t '['
# or
# '<,'>! LC_ALL=C sort -k 2 -t '['

     FARM([A], [Ani kůň netáhne víc než může.])
     FARM([A], [Ani slepice darmo nehrabe.])
   AUGUST([B], [Bylo-li léto mokré, bývá v příštím r»
     FARM([B], [Bitím podaří se jednoho čerta vyhnat»
    MARCH([B], [Březen bez vody, duben bez trávy.])
    MARCH([B], [Březen, za kamna vlezem; duben, ešť»
     FARM([C], [Co doma uvaříš, to doma sněz.])
     FARM([C], [Cvičení a zkoušení dává umění.])
 FEBRUARY([C], [Co do hromnic, to od hromnic, spíše »
 FEBRUARY([C], [Co si únor zazelená, březen si to há»
      MAY([C], [Co červ nesní, to sud polkne (květ v»
 NOVEMBER([C], [Chodí-li husa o sv. Martině po ledě,»
   AUGUST([Č], [Čeho červenec a srpen neuvaří, toho »
     FARM([Č], [Čas růže přináší.])
     FARM([Č], [Častá dohlídka pohání práci.])
     JUNE([Č], [Červen studený, sedlák krčí rameny.])
   SPRING([Č], [Čemu podzim říká: hnilo! tomu jaro p»
    APRIL([D], [Duben hojný vodou, říjen pivem.])
 DECEMBER([D], [Dvanáct nocí od štědrého večera až d»
     FARM([D], [Dá-li Pán Bůh kravičku, dá též travi»
     FARM([D], [Dvojí orání, dvojí chléb, trojí orán»
  JANUARY([D], [Dobrá sanice, koňská smrt.])
     JULY([D], [Den sedmi bratrů když deštivý, bývá »
     JUNE([D], [Dlouhý den, krátká nit.])       <---»
     JUNE([D], [Do sv. Jana Křtitele nechval ječmene»
      MAY([D], [Déší na Mamerta přináší soužení (11.»
      MAY([D], [Déšť v noci před sv. Valburgou naplň»
   SUMMER([D], [Deštivé léto horší podzimku.])
     FARM([H], [Hrne se do toho, jako vůl do kupy se»
     FARM([H], [Hubu zapni a oči napni.])
      MAY([H], [Havlovo žito, Urbanův oves, co z toh»
 NOVEMBER([H], [Hřímá-li v listopadu, bude dobrý rok»
     FARM([I], [I slepá slepice někdy zrnko najde.])
    APRIL([J], [Jak dlouho žába před sv. Jiřím křičí»
    APRIL([J], [Jiří a Marek mrazem nás zalek'.])
 DECEMBER([J], [Jaké počasí na sv. Barboru, takové b»
     FARM([J], [Jaká řepa, taká nať, jaké plášti, ta»
     FARM([J], [Jsou-li v zimě rampouchy hojné, budo»
 FEBRUARY([J], [Je-li v únoru zima a sucho, bývá prý»
  JANUARY([J], [Jaké počasí je který den od vánočníh»
  JANUARY([J], [Jména Ježíš, ke kamnům nejblíž.])
     JUNE([J], [Jaká povětrnost jest na sv. Jana Křt»
     JUNE([J], [Jaké počasí na Ladislava se ukazuje »
    MARCH([J], [Jak jest na popeleční středu, tak bu»
    MARCH([J], [Jedna vlašťovka nedělá léto.])
      MAY([J], [Je-li už máj zahradníkem, není stodo»
 NOVEMBER([J], [Jaký den sv. Kateřiny, takové počasí»
 NOVEMBER([J], [Je-li zamračeno na sv. Martina, bývá»
installfest_2019.mc                    1,1          Top
```

```
define([BOOK_NAME], [<b>]defn([BOOK_NAME])[</b>])

# description
define([___DESCR], [define([DESCRIPTION], [<p>$1</p»

# description,  append if duplicit
#define([___DESCR], [define([DESCRIPTION], defn([DE»
#])])

# define auxiliary symbol for navigation
define(.LANG_CODE)

m4wrap([
        divert(0)dnl
<!DOCTYPE html>
<!-- DO_NOT_EDIT -->
<html>
<head>
<meta charset="UTF-8">
<title>AUTHOR</title>
<style>
html {
        font-size: 120%;
}
section ul li {
        display: inline;
}
</style>
</head>
<body>
<header>
dnl[]include([lang.nav])dnl     <--- do not hardcod»
include(NAVIG_FILE)dnl
dnl[]sinclude(NAVIG_FILE)dnl    <--- silent include»
dnl[]undivert(NAVIG_FILE)dnl    <--- no macro expan»
<h1>AUTHOR</h1>
DESCRIPTION
</header>
<nav>
        <h2>MONTHS</h2>
        <ul>
divert(AFTER_MONTHS)dnl
        </ul>
        <h2>SEASONS</h2>
        <ul>
divert(AFTER_SEASONS)dnl
        </ul>
        <h2>THE_OTHERS</h2>
        <ul>
divert(CLOSE_NAVIG)dnl
        </ul>
</nav>
<article>
divert(LAST_BUFFER)dnl
</article>
<footer>
        <p><a href="PROJECT_ROOT_URL">PROJECT_NAME<»
</footer>
</body>
</html>
])
html_proverbs.m4                       89,0-1       Bot
```

```
<!DOCTYPE html>
<!-- DO NOT EDIT! Generated automatically! -->
<html>
<head>
<meta charset="UTF-8">
<title>Karel Hrubý</title>
<style>
html {
        font-size: 120%;
}
section ul li {
        display: inline;
}
</style>
</head>
<body>
<header>
<a href="html_proverbs_cz.html">Czech</a> (cz) |
<a href="html_proverbs_sk.html">Slovak</a> (sk) |
<b style="color:red">English</b> (en) |
<a href="html_proverbs_pl.html">Polish</a> (pl) |
<a href="html_proverbs_fi.html">Finnish</a> (fi) |
<a href="html_proverbs_hu.html">Hungarian</a> (hu) |
<a href="html_proverbs_de.html">German</a> (de) |
<a href="html_proverbs_ru.html">Russian</a> (ru) |
<a href="html_proverbs_uk.html">Ukrainian</a> (uk) |
<a href="html_proverbs_fr.html">French</a> (fr) |
<a href="html_proverbs_it.html">Italian</a> (it) |
<a href="html_proverbs_hi.html">Hindi</a> (hi) |
<a href="html_proverbs_mn.html">Mongolian</a> (mn) |
<a href="html_proverbs_ja.html">Japanese</a> (ja) |
<a href="html_proverbs_zh_CN.html">Chinese (Simplif»
<a href="html_proverbs_so.html">Somalia</a> (so) |
<a href="html_proverbs_sw.html">Swahili</a> (sw) |
<a href="html_proverbs_yi.html">Yiddish</a> (yi) |
<a href="html_proverbs_fa.html">Persian</a> (fa) |
<h1>Karel Hrubý</h1>
<p>HRUBÝ, Karel. <b>A collection of proverbs, sayin»
</header>
<nav>
        <h2>Months</h2>
        <ul>
                <li><a href="#jan">January</a></li»
                <li><a href="#feb">February</a></li»
                <li><a href="#mar">March</a></li>  »
                <li><a href="#apr">April</a></li>  »
                <li><a href="#may">May</a></li> <!-»
                <li><a href="#jun">June</a></li>   »
                <li><a href="#jul">July</a></li>   »
                <li><a href="#aug">August</a></li> »
                <li><a href="#sep">September</a></l»
                <li><a href="#oct">October</a></li»
                <li><a href="#nov">November</a></li»
                <li><a href="#dec">December</a></li»
        </ul>
        <h2>Period</h2>
        <ul>
                <li><a href="#spr">The Spring</a></»
                <li><a href="#sum">Summer</a></li> »
                <li><a href="#aut">Autumn</a></li> »
                <li><a href="#wint">Winter</a></li»
html_proverbs_en.html                  1,1          Top
```

Vim abbreviations & placeholders

```
$ cat ~/.vim/ftplugin/m4/m4_maps.vim

…
iabbrev <buffer> de define([], [%%%])<c-o>T(<right><c-r>=Eatchar('.')<cr>
iabbrev <buffer> def define([], [<cr><tab>%%%<cr><c-d>])<up><up><c-o>f]<c-
r>=Eatchar('.')<cr>
…

de → define([], [%%%])

def → define([], [
        %%%
])
```

Debugging

- Output redirected to **debug.m4**
- Only appends data
  - More debug places in m4 code
- Use Makefile to truncate **debug.m4** file!

```
$ > debug.m4

$ cat script.m4
…
debugfile([debug.m4])debugmode([taeq])
BUGGY([…])
debugmode
…
```

Stacks

- Unlimited number
- Unlimited size
- Nested source code generation
  - Transition to the next node

```
# β
define([NODE1], […  pushdef([$0], defn([NODE2])) …  popdef([$0]) …])
define([NODE2], […  popdef([$0]) …])

# A → β
define([NESTED], [
    …
    pushdef([$0], defn([NODE1]))
    …
])
```

Trapped by **ifelse(...)**

NOT FOR GENERAL PURPOSE BRANCHING!

1. Small branches only
2. End condition of a loop
3. Automaton (transition/evaluation)
4. Comments **ifelse([... ignored ...])**

Branching in M4

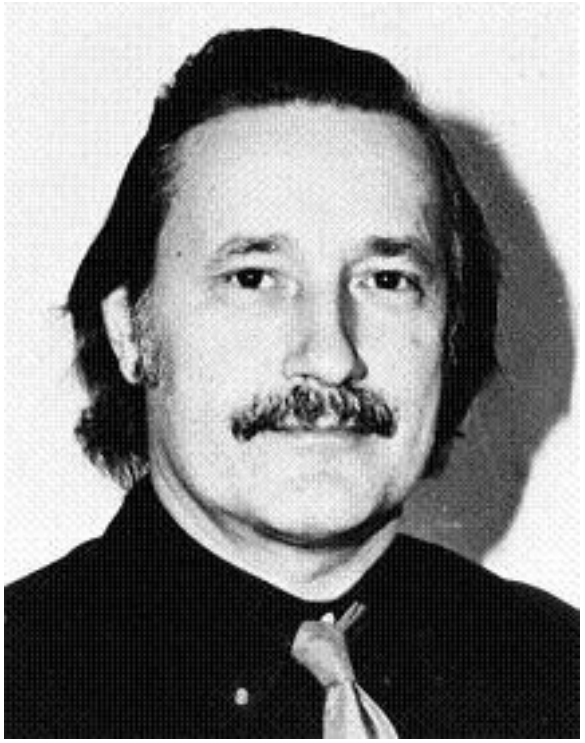
```
define([A_0], [zero])
define([A_1], [one])
define([A_2], [two])
define([A_3], [three])
define([A_4], [four])
define([A_5], [five])
define([A_6], [six])
define([A_7], [seven])
define([A_8], [eight])
define([A_9], [nine])
define([A], [A_$1])
A(3)
three
#because: A(3) → A_$1 → A_3 → three
```

A brief history of M4

- **GPM** (**G**eneral **P**urpose **M**acrogenerator)
  - written by **Christopher Strachey**
  - GPM fit into 250 machine instructions!
- **M3** written by **Dennis Ritchie** for the AP-3 minicomputer
- **M4** for the original versions of UNIX, written by
  - **Dennis Ritchie**
  - **Brian Kernighan**
  - Jim E. Weythman
    ```
    divert(-1)
    define(…)
    …
    divert
    ```
  - Rick Becker
  - John Chambers
  - Doug McIlroy
  - …

Authors of M4

GPM (1965)

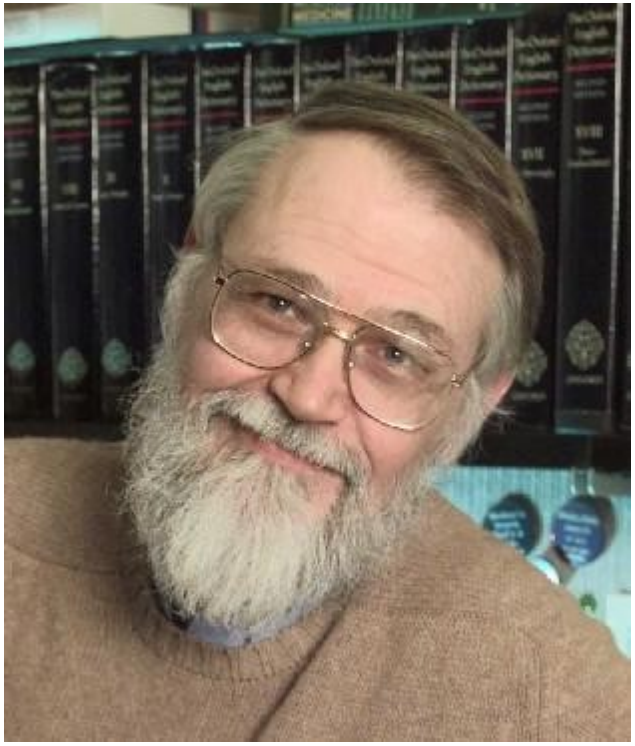


Christopher Strachey

C (1973), M3, M4 (1977)

Dennis Ritchie

C, M4, AWK

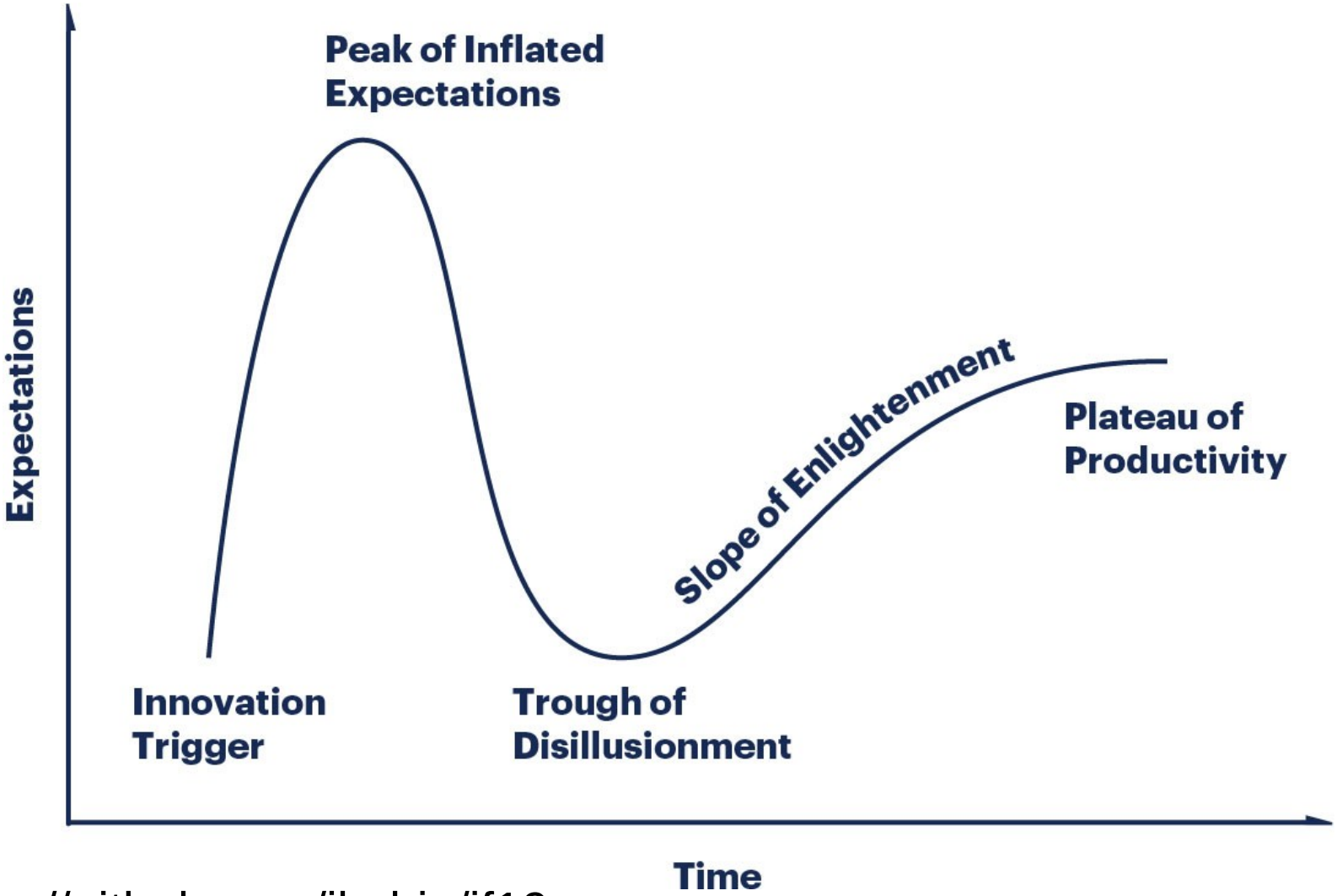


Brian Kernighan

and several other authors ...

Hype cycle

- Angry at the M4? Keep calm and carry on!



References

- https://www.gnu.org/software/m4/manual/m4.html
- https://www.gartner.com/en/research/methodologies/gartner-hype-cycle
- https://www.cs.cmu.edu/~mihaib/kernighan-interview/
- https://pc.zoznam.sk/novinka/zomrel-tvorca-unixu-jazyka-c
- https://www.computerhope.com/people/christopher_strachey.htm

Děkuji za pozornost!

Nezapomeňte vyplnit anketu!

InstallFest 2019
Josef Kubín