md2review で Markdown から Re:VIEW に変換すると箇条書きの入れ子は無視される

※ 2014 年 7 月頭の状況です。

Markdown と Re:VIEW を知っている方しか見ない記事だと思うので細かい説明は抜きでいきます。

ソフトウェア技術者だと最近は Markdown で資料を書くことが多いのではないかと思います。私もそうなのですが、Markdown で書いたお資料を印刷する際に困っていました (客様に渡す報告書等)。

いろいろ試した結果 Mou(Mac 用の Markdown エディター) で印刷していたのですが、フォントが中国系になるという不満点がありました。

そこで md2review で Markdown から Re:VIEW に変換して PDF 化するというフローを試したところかなりいい感じに出力できたのですが、私にとっての大問題が判明しました。

そう、「**md2review で Markdown から Re:VIEW に変換すると箇条書きの入れ子は無視される**」のです。

番号付き箇条書き (Decimal 型) については Re:VIEW 側が対応していないのでしょうがないのですが (個人的にあまり使ってないですし)、普通の箇条書き (Disc 型) の入れ子は多用しているので大変厳しい。

そこで、改造して Pull Request 出したろう！ と着手したのですが、

- md2review 調査
  - かなりの部分 redcarpet 依存であると判明。
- redcarpet 調査
  - 「メイン処理は C で書かれてるのかよ〜」と思いつつ読んだ結果、簡単には対応できそうもないと判断。なぜ md2review が箇条書きの入れ子に未対応なのか、納得する。
- 残念対応
  - md2review で変換した Re:VIEW の箇条書き部分に入れ子の状態から挿入される空行に規則性があるようだったので、これを解析したらどうにかならないか？ と思う。
  - redcarpet に postprocess というレンダリングの後処理を行うための仕組みがあったのでそこで処理してみた。

- 作ってみた結果、以下の問題がある残念なものが出来上がった。
  - 入れ子 2 段までしか判断できないため 3 段以上は正常に動作しない。
  - 箇条書きの最後のアイテムについては判断できないためとりあえずワーニングコメント#@warn(CONFIRM NEST!) を出力する (確認しやすくはなった)。

という残念な結果に終わりました。

残念対応ですが無いよりはマシですので公開しておきます。

md2review の review.rb に以下の postprocess を追加します。

```
def postprocess(full_document)
  # レンダリング結果を解析して順番なし箇条書きのインデントを反映する。
  # ただし 2 段までしか判断できないため 3 段以上は正常に動作しない。
  # また、箇条書きの最後のアイテムについては判断できないため
  # ワーニングコメント#@warn(CONFIRM NEST!) を出力する。

  # 無効化するときは次の行の return を有効化する。
  #return full_document
  require 'set'

  lf_del_set = Set.new
  lf_ins_set = Set.new

  pre_ul_flg = false
  pre_emp_flg = false
  ul_level = 1

  lines = []
  full_document.split(/\n/).each_with_index do |line, i|
    if line == "" || line[0, 3] == " * "
      if line == ""
        if pre_emp_flg && pre_ul_flg
            line = "\#@warn(CONFIRM NEST!)\n"
        elsif pre_ul_flg
          lf_del_set.add(i)
          if 1 < ul_level
            if lines[i - 1][0, 2] == " *" && lines[i - 2][0, 2] == " *"
              lines[i - 1] = " #{'*' * (ul_level - 1)} " +
                lines[i - 1][(ul_level + 2), lines[i - 1].length]
              ul_level -= 1
              lf_ins_set.add(i - 1)
            end
          end
        end

        pre_emp_flg = true
      else
        if pre_ul_flg && pre_emp_flg
          ul_level += 1
        end

        line = " #{'*' * ul_level} " + line[3, line.length]

        pre_ul_flg = true
```

```
        pre_emp_flg = false
      end
    else
      pre_ul_flg = false
      ul_level = 1
    end

    lines.push(line)
  end
  if pre_ul_flg
    lines.push("\#@warn(CONFIRM NEST!)\n")
  end

  ret_document = ""
  lines.each_with_index do |line, i|
    if lf_ins_set.include?(i)
      ret_document += "\n"
    end
    if !lf_del_set.include?(i)
      ret_document += line + "\n"
    end
  end
  ret_document
end
```