テキスト処理 第8回 (2006-06-13)
行末パターン実装レポート説明

田中哲
産業技術総合研究所
情報技術研究部門
akr@isc.senshu-u.ac.jp
http://staff.aist.go.jp/tanaka-akira/textprocess/

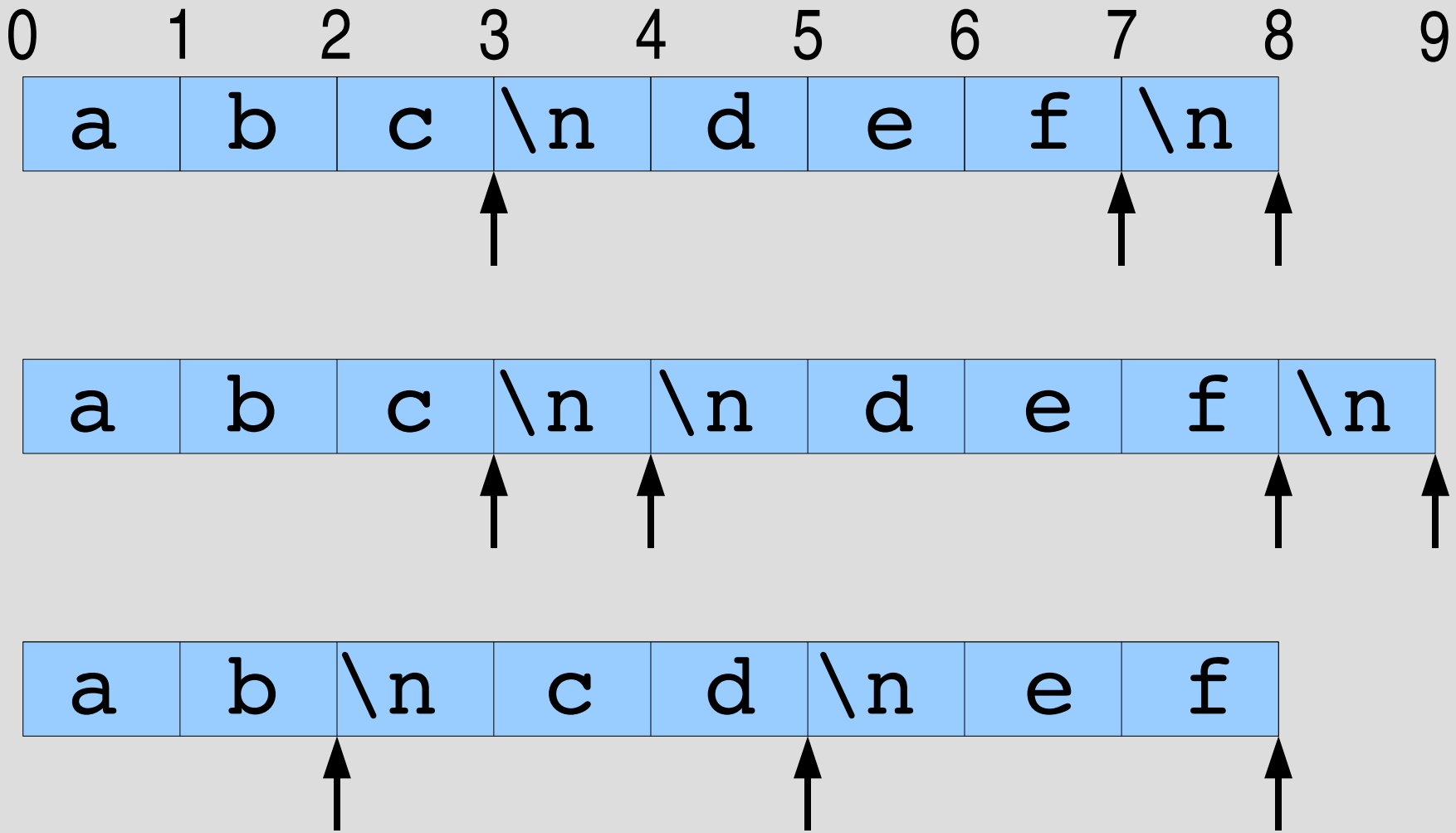
レポート

- 行末にマッチするパターン [:line_end] を実装して解説せよ
- ユニットテストを提供するので、実装したらテストして確認すること
- ∠切 2006-06-13 16:20
- IT's class
- 拡張子が txt なテキストファイルを望む

行末 $

- 行の末尾にマッチするパターン [:line_end]
- Ruby の正規表現では $
- このパターン自身は文字を消費しない
- 文字列の最後と改行の直前にマッチする
- 文字列が改行で終わっていても文字列の最後にマッチするのは歴史的慣習

行末

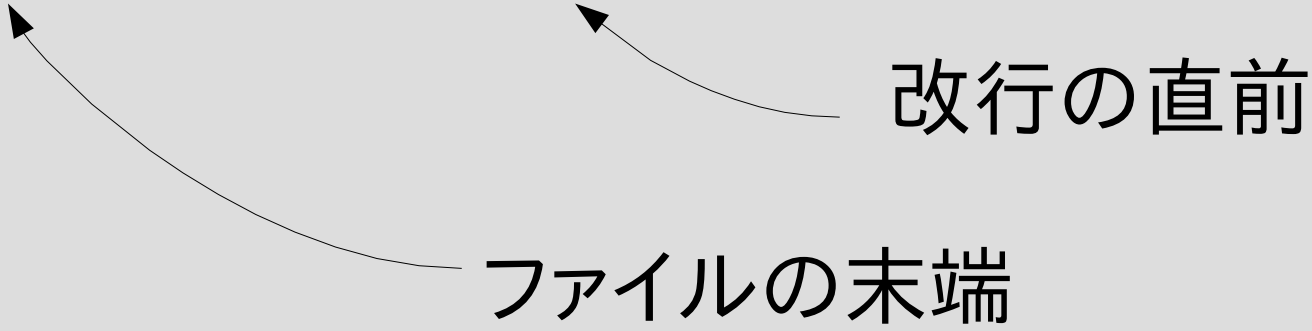


実装: try

```
def try(exp, seq, pos, &block)
  ...
  when :line_end
    try_line_end(seq, pos, &block)
  ...
end
```

実装: try_line_end

```
def try_line_end(seq, pos)
  if pos == seq.length || seq[pos] == "\n"
    yield pos
  end
end
```



pos < seq.length

```
def try_line_end(seq, pos)
  if pos == seq.length || (pos < seq.length && seq[pos] == "\n")
    yield pos
  end
end
```

seq[pos] が範囲内である条件

害はない

pos == seq.length なら到達しないので常に真

範囲を考慮している、という意図の表明くらい

\r 対応

```
def try_line_end(seq, pos)
  if pos == seq.length ||
     (seq[pos] == "\n" || seq[pos] == "\r")
    yield pos
  end
end
```

変換してあればいらないんだけど、
変換しないで読み込むこともある

ざっと見た結果

- ほとんどのひとは問題ないコードが書けている
- yield pos+1 としている人がいた
  それだと文字を消費してしまう
  (ユニットテストも失敗する)
- わからない場合、どこがわからないのか書いてあればなるべく対応する