テキスト処理 第12回 (2008-07-08)
scanstr レポート解説

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-2008/

レポート

- scanstr(s, r) を実装して解説せよ

- 実装したらユニットテストで確認してほしい
- 〆切 2008-07-08 12:00
- RENANDI
- 拡張子が txt なテキストファイルがよい

scanstr(s, r)

- String#scan を部分的に真似したもの
- 文字列 s 中の r にマッチする部分文字列をすべて調べ、配列として返す

- 実行例
- p scanstr("banana", [:anychar])
  #=> ["b", "a", "n", "a", "n", "a"]
- p scanstr("banana", [:alt, "b", "n"])
  #=> ["b", "n", "n"]
- p scanstr("banana", [:cat, "n", "a"])
  #=> ["na", "na"]

scanstr の実装

```
def scanstr(s, r)
  s = s.split(//)
  beg = 0
  result = []
  while md = cap_include(r, s, beg)
    m = md[:all].begin
    n = md[:all].end
    result << s[m...n].join
    beg = n
    if m == n
      if beg == s.length
        break
      else
        beg += 1
      end
    end
  end
  result
end
```

実装方針

- だいたい gsubst と同じ
- マッチを見つけたら result に追加していく

gsubst

```
def gsubst(s, r)
  s = s.split(//)
  beg = 0
  result = []
  while md = cap_include(r, s, beg)
    m = md[:all].begin
    n = md[:all].end
    h = {}
    md.each {|k,v| h[k] = s[v].join }
    result += s[beg...m]
    result << yield(s[m...n].join, h)
    beg = n
    if m == n
      if beg == s.length
        break
      else
        result << s[beg]
        beg += 1
      end
    end
  end
  result += s[beg..-1]
  result.join
end
```

gsubst から scanstr へ

```
def scanstr(s, r)
  s = s.split(//)
  beg = 0
  result = []
  while md = cap_include(r, s, beg)
    m = md[:all].begin
    n = md[:all].end
    h = {}
    md.each {|k,v| h[k] = s[v].join }
    result += s[beg...m]
    result << yield(s[m...n].join, h)
    beg = n
  end
  result << s[beg...s.length]
end
```

マッチ情報の準備
マッチ間の追加

ブロック呼出し

```
    if m == n
      if beg == s.length
        break
      else
        beg += 1
      end
    end
  end
```

スキップ部分の追加

最後のマッチ後を追加

連結

ざっと眺めた結果

- 簡単だった模様

- gsubst を「使った」実装
  - res = []; gsubst(s, r) {|str, h| res << str }; res
  - たしかに動く
- gsubst を改造して、h を取り除いていないもの
  - 動くけど無駄
- 無限ループ防止が入っていないもの