



分散ハノイプロトコル(GDHP)と国際交流

g新部 裕

NIIBE Yutaka <gniibe@fsij.org>

特定非営利活動法人 フリーソフトウェアイニシアティブ

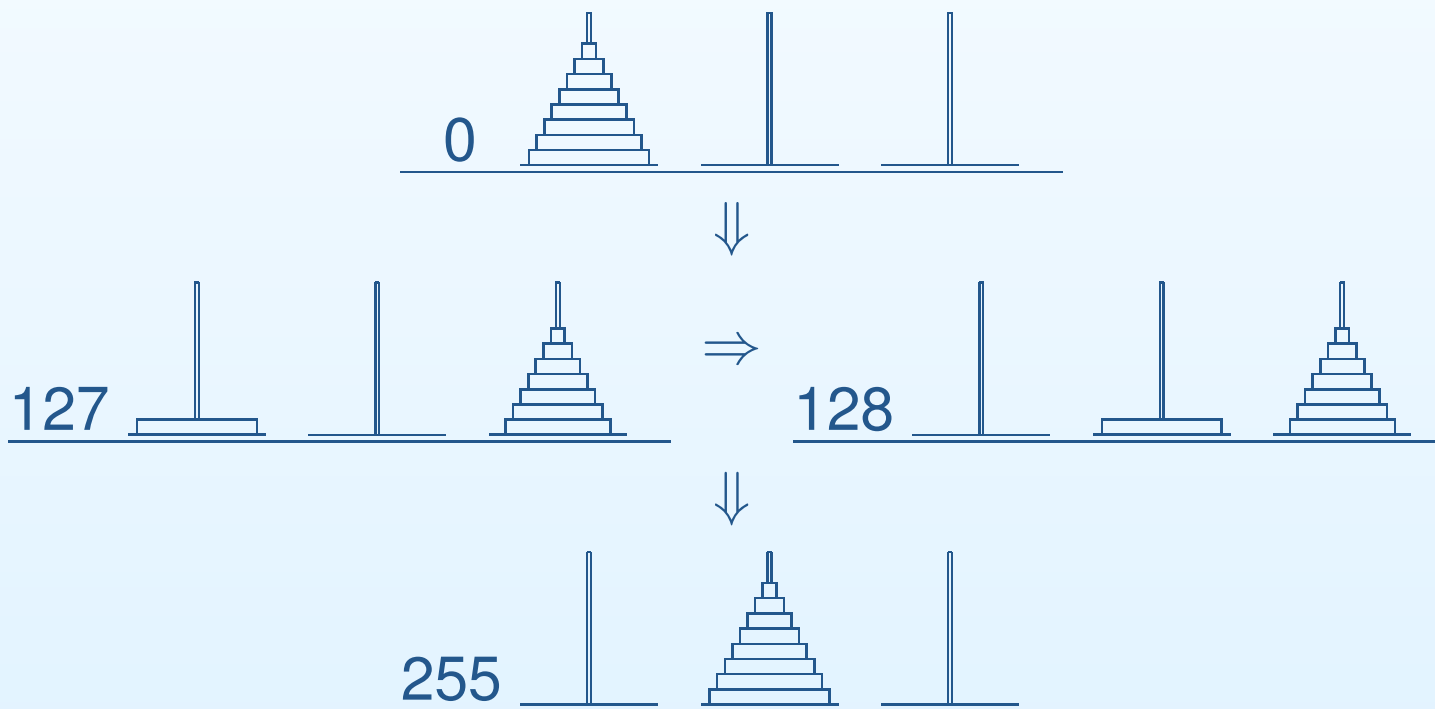


もくじ

- ハノイの塔
- 分散ハノイプロトコル
- POSIX Thread版 実装
- TCP/IP (GTK+)版 実装
- 人間による実装 (国際交流)
- まとめ

ハノイの塔

- 19世紀にÉdouard Lucas (ペンネームProfessor N. CLAUS) により発明された数学の問題.
- 再帰計算のプログラミングの例題によく使われる.
- N 枚の円盤の場合, $2^N - 1$ 手



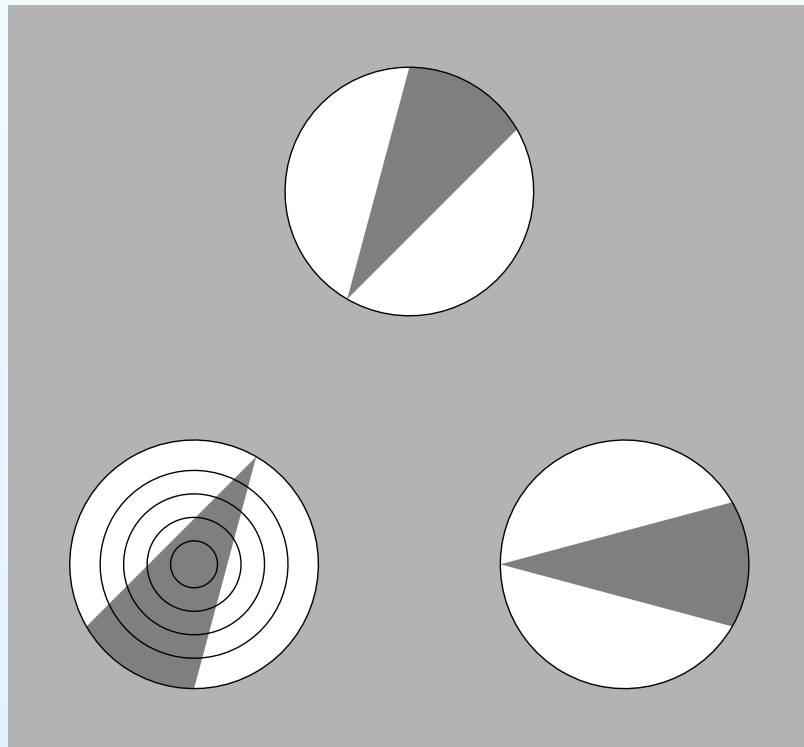
ハノイの塔 (続き)

- Scheme で記述すると下記の通り:

```
(define (toh-move N from to aux)
  (cond ((= N 1) (print (format "move ~A to ~A" from to))
        (else (begin (toh-move (- N 1) from aux to)
                      (toh-move 1 from to aux)
                      (toh-move (- N 1) aux to from))))))
(toh-move 8 'TowerA 'TowerB 'TowerC)
```

分散ハノイプロトコル (GDHP)

- “g” Distributed Hanoi Protocol
- 3つのタスクが, それぞれひとつの塔に対応する.
- タスクは2つの状態(向き)を持つ.
- 左を向いているか右を向いているか.



GDHP(初期状態)

- 塔のうちひとつは移動元であり, 全ての円盤はこの塔に(順番に)乗っている。
- 塔のうちのひとつは移動先である。
- 塔のうちのひとつは補助である。
- 塔の向き
 - 移動先の塔は移動元を向く。補助の塔も移動元を向く。
 - 移動元の塔の向きは, 円盤の数の奇偶によって, 奇数なら移動先, 偶数なら補助の塔を向く。

GDHP(プロトコル)

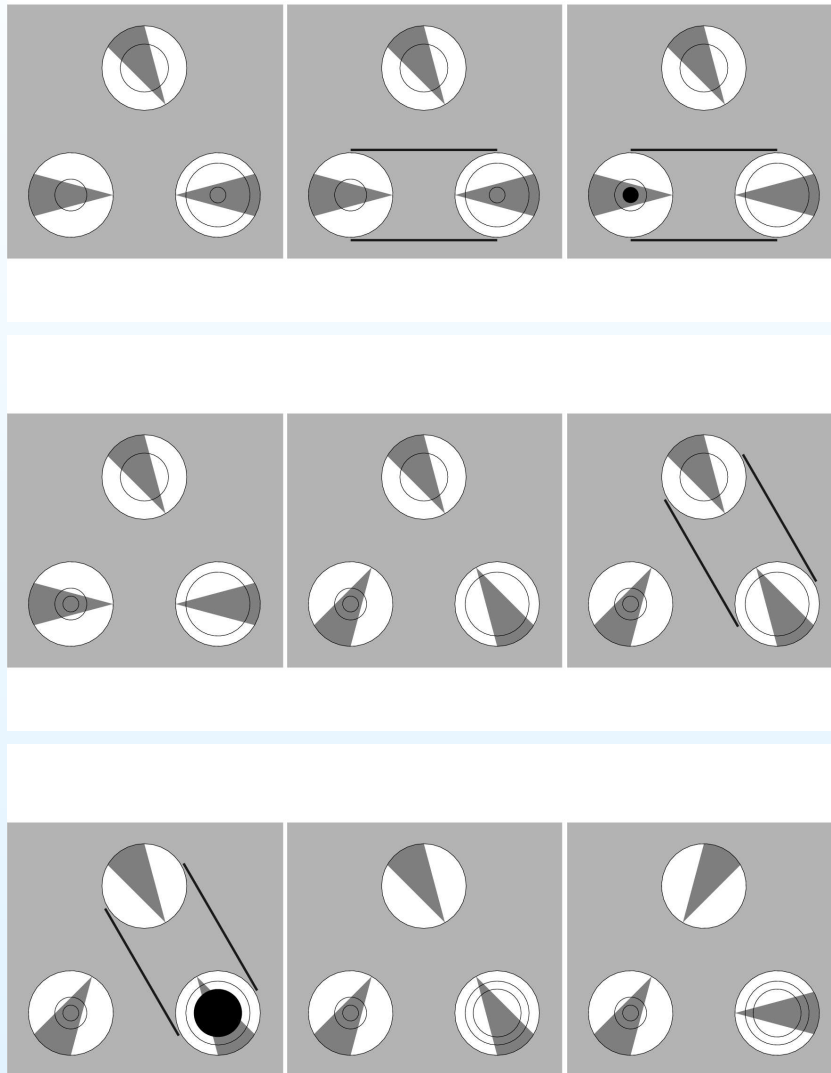
下記のやりとりを終了条件となるまで繰り返す:

- 向き合った同士のタスク(塔)が円盤をやりとりする。
 - 一番上の円盤の大小を比較する(円盤がない場合は ∞ とする).
 - 大きな円盤を持っているタスク(塔)が勝ち, 負けた方から円盤を受け取り, 一番上に置く.
 - 円盤のやりとりをしたタスクが向きを変える.

GDHP(終了条件)

- 全ての円盤が移動先に移れば終了。
 - 向き合ったタスク(塔)の両方が円盤がない状態となり, 円盤のやりとりが行われなくなる(デッドロック)。

GDHP(図解)



POSIX Thread版 実装

- 塔をひとつのスレッドとし, POSIX Thread でC 言語で実装.
- <http://www.gniibe.org/code/towers-of-hanoi.c>
- ルーチン
 - do_hanoi
 - exchange
 - synchronize

TCP/IP (GTK+)版 実装

- 塔をひとつのプロセスとし, ユーザインタフェースとしてGTK+を用いる.
- TCP/IP の通信によってプロトコルを実装.
- <http://www.gniibe.org/code/gtk-toh.scm>

人間による実装 (国際交流)

- ハノイ (March 2004)
- 東京 (November 2004)
- ソウル (December 2004)
- 北京 (March 2005)

まとめ

- 塔に対応した**3**つの並行タスクによる「ハノイの塔」.
- **3**人よれば文殊の知恵.
- 自由ソフトウェアのハッカーをして、**独立自尊と協調の大切さを認識**.
- **みんなで体得しよう!**

Happy Hacking!