

分散ハノイプロトコル (GDHP) と国際交流

g新部 裕

<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

2005 年 6 月 3 日

要旨

本論文では、パズル「ハノイの塔」を解く、分散ハノイプロトコル (“g” Distributed Hanoi Protocol, 以下 GDHP) について述べる。GDHP は塔に対応する 3 つの異なるタスクがそれぞれ通信を行い、協調して問題を解くプロトコルである。各タスクは部分的な情報を持つのみで、単純な手順によって円盤を交換する。

まず GDHP について解説し、次に GDHP を実装したプログラム 2 つ (POSIX Thread 版, TCP/IP 版) について説明する。また、GDHP が自由ソフトウェアハッカーの相互の親睦を深めるツールとなり国際交流に有効であることを示す。

1 はじめに

「ハノイの塔」は、19 世紀に Édouard Lucas により (ペンネーム Professor N. CLAUS として) 発明された数学の問題であり [1], 再帰計算のプログラミングの題材として利用されることの多いパズルである。

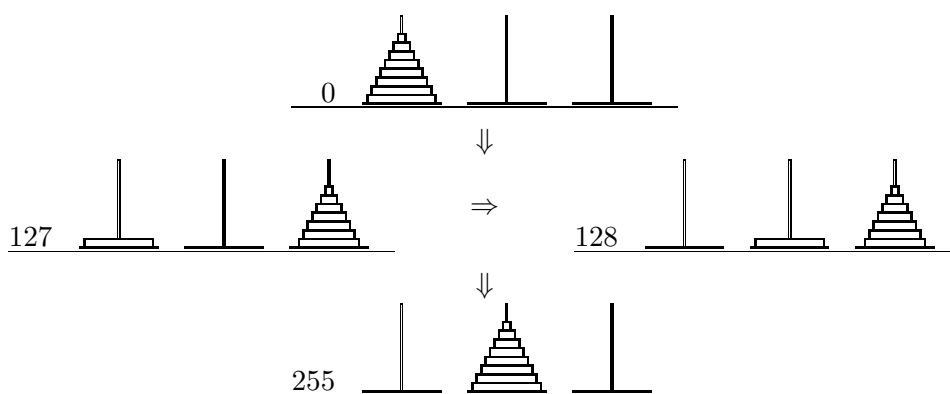


図 1: ハノイの塔

図1ではハノイの塔の初期状態(0手目), 手数のもんなかあたり(127手目, 128手目), および最終状態(255手目)を示す。¹

パズルでは, 8枚の大きさの異なる円盤と3つの塔があり, 初期状態ではひとつの塔(移動元)に全ての円盤が大きい順に乗っている。必ず大きい円盤の上に小さい円盤が乗るという制約を保ち, 第三の補助塔を利用しつつ, 移動先の塔に全ての円盤を移動するのがパズルの目的である。

一枚の円盤の移動を一手と数えると, 8枚の円盤を移動させる最小手数は255手となる(N 枚の円盤の場合, $2^N - 1$ 手)。

解法を出力として得るプログラムを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)
```

再帰を利用すると, 移動元の塔から補助塔への $N - 1$ 枚の円盤の移動, 一番大きい円盤の移動(移動元から移動先へ), そして補助塔から移動先の塔への $N - 1$ 枚の円盤の移動と記述することができ, それで解法が得られる。

このように単純なハノイの塔の問題と美しいその解法であるが, 2003年にハノイを訪問したことを契機として, (あえて)3つのタスクでハノイの塔を解くということに挑戦してみたところ, これまた面白いプロトコルとなった。

以下では, このプロトコルと, ソフトウェア, そして国際交流への応用について述べる。

2 GDHP(“g” Distributed Hanoi Protocol)

3つのタスクがあり, それぞれひとつの塔に対応するとする。図2に上から見たハノイの塔を示す。この図では円盤は4枚としてある。

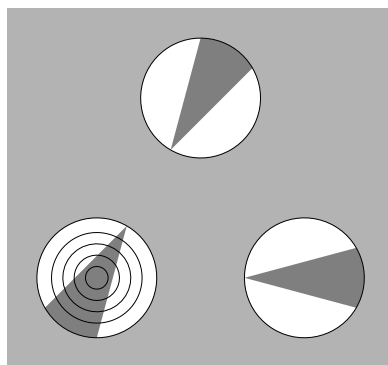


図2: ハノイの塔を上から見たところ(4枚の円盤)

¹ハノイの塔を拡張して伝教と結び付け, 64枚の円盤の移動が終わった時が世界の終末だとする伝説は「梵天(Brahma)の塔」と呼ばれる。

GDHP は下記の通りであり, この手順でパズルが解ける。

タスク (塔) の取る状態とやりとり

- 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。

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。
- 塔のうちひとつは移動先である。
- 塔のうち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。

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。

終了条件とその検出

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)。

図 3: GDHP(“g” Distributed Hanoi Protocol)

具体例として, GDHP によって円盤のやりとりが行われて進展する状況を図 4に示す。タスク (塔) が二つの向きを持ち, 向き合った同士が円盤をやりとりする。大きい円盤を持っている方が小さい円盤を受け取り上に置く。やりとりが終わったら向きを変える, というようにプロトコルが進む。

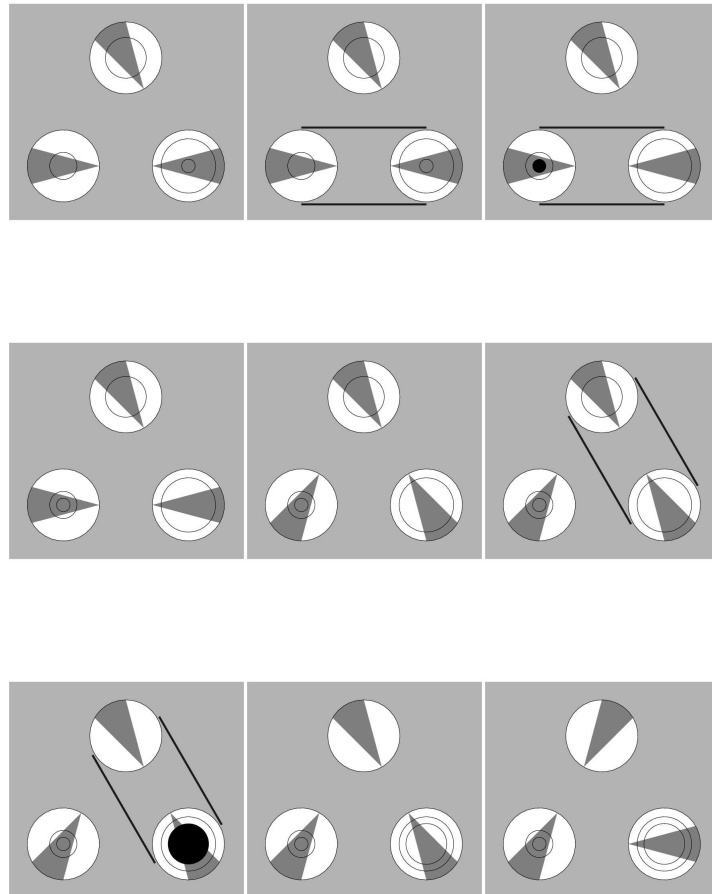


図 4: GDHP の図解

3 GDHP の実装

GDHP の実装として二つのプログラムを書いた。本章ではこの二つのプログラムについて述べる。

3.1 POSIX Thread での実装

塔をひとつのスレッドとし、3つのスレッドで円盤をやりとりするプログラムを POSIX Thread で C 言語で実装した。

ひとつのファイル `towers-of-hanoi.c` で実現され、POSIX Thread ライブラリとリンクして実行ファイルは作成される。

円盤の移動を標準出力に `printf` により出力するとした。

コマンドライン引数として円盤の枚数 N を取り、狭義のハノイの塔 (8 枚) だけでなく、一般の円盤の数に対応している。

メインルーチンで円盤の枚数により方向を決め、3つのスレッドを生成し、各スレッドは、ルーチン `do_hanoi` で GDHP を繰り返す。

ルーチン `do_hanoi` は二つのルーチン呼び出す。円盤のやりとりのためのルーチン `exchange` と、出力の順番の整合のためとなるの塔と同期するルーチン `synchronize` である。`synchronize` が無い実装では、計算は正しく行われるが出力が前後することがあり得るためである。

3.2 TCP/IP での実装

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+ を使い、3つのプロセスで TCP/IP によってプロトコルを実装し円盤をやりとりするプログラムを Scheme インタプリタの `Gauche` で実装した。

ひとつのファイル `gtk-toh.scm` で実現される。

円盤の移動を GTK+ のグラフィカルユーザインタフェースでアニメーションとして見せることとした。

プログラムはサーバーモード、クライアントモードあるいは両方のモードで起動することができる。サーバーモードはソケットを `listen` し、クライアントからの GDHP を受け付けるモードである。クライアントモードはこちらからサーバーにつながるモードである。両方のモードとは、一台のホストで3つのプロセスを稼働させるモードである。サーバーとクライアントは TCP/IP のネットワークで通信できれば良く、別々の3台のホストで実行できる。

通信プロトコルは、3つのプロセスの通信の確立、GDHP の実行、および通信の終了時の処理の3つからなる。純粋な GDHP の他、実行の中止にも対応している。

4 人間による GDHP

GDHP は単純なプロトコルであり、人間でもちょっとした訓練で習得できる。プロトコルそのものは、ハノイの塔を一人で解くよりも簡単に理解しやすい。言葉でプロトコルを説明すると複雑に聞こえるが、体を動かして実践とともに習得を進めると理解が容易である。

ひとり一人をタスク (塔) とし、3人で三角形を構成するように並ぶ。円盤は大きさが違うものは色も変えて区別しやすいようにするのが望ましい。

気を付ける点は、「円盤を交換した後に向きを変える」ということである。自分以外の二人が向きを変える際、つられて動きやすいので注意する。

これまで、ハノイ (2004/03), 東京 (2004/11), ソウル (2004/12), 北京 (2005/03) とアジアの各都市で自由ソフトウェアハッカーを対象として、GDHP の訓練を実施してきた。

最初はハノイにおいて、第三回アジア OSS シンポジウムの BoF として実施した。POSIX Thread での実装および TCP/IP での実装を議論したが、説明のために人間によるもの (円盤には、洗顔用石鹸、コースター、CD-ROM (小さい順)、マウスパッドを利用した) が最も面白いことが発見できた。

次に東京において、ダンボールに色紙をつけた円盤を作成し、これを用いて実施した。続いて、ソウル、北京 (長城) でもこの円盤を用いて訓練を行った。

特にソウルでは、韓国の自由ソフトウェアハッカーのイベント (第三回 Codefest) において解説を行い、参加者 40 名程度の皆で GDHP を行い、親睦を深めることができた。

プログラミングについて勉強したことがあれば、ハノイの塔のバズルは大抵の場合知っている。GDHP はそのハノイの塔の別の解法であり、3 人で協力することによって簡単に解くことができるというデモンストレーションとなる。

これは、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し、また、協力によってなされる成果を再確認するというとても良い効果がある。

同様のハッカー間の親睦を深めるツールとしては、GPG の鍵交換がある。鍵交換の際、お互いの名前を確認し、自己紹介をする慣習が自由ソフトウェアの間、特に Debian では良く行われる。しかし、GPG の鍵交換については、GPG が必ずしも常に使うツールではないことから、初心者にとって参加しにくいという問題があった。

GDHP は自由ソフトウェアハッカーの誰もが参加でき、知りあいの二人がもう一人を紹介するという形で、自由ソフトウェアの集団を強固にし発展させることができる。GPG の鍵交換を補完し、親睦を深めるツールとして展開が期待できる。

5 おわりに

ハノイの塔は、20 年ほど前までにはいくつかの論文が出されているが、最近では見られない。

GDHP の塔にタスクを対応させるという方式はおそらくオリジナルであろうと考えられ、それは、なによりも簡潔で面白いプロトコルである。

最後に、ここで GDHP の研究開発に協力いただいた各人に謝辞を述べたい。

GDHP の着想と初めての実演に応援をいただいた、第三回アジア OSS シンポジウム “ハノイの塔 BoF” の参加者各位、および (財) 国際情報化協力センターのみなさんに感謝する。(独) 産業技術総合研究所、つくばのみんなありがとう。

GDHP を作る前のいくつかの挑戦、中でも www.gniibe.org の CGI スクリプトの実装に協力いただいた田中 哲さん、また、Gauche の作者である川合史郎さんに感謝する。彼ら無くして、GDHP は生まれなかった。

Happy Hacking.

参考文献

- [1] Édouard Lucas, “Récréations Mathématiques”, vol.III., Gauthier-Villars (Paris) 1893.

- [2] NIIBE Yutaka, “Towers of Hanoi”, <http://www.gniibe.org/>, 2003-2005.
- [3] NIIBE Yutaka, “Towers of Hanoi by Threads”, <http://www.gniibe.org/code/towers-of-hanoi.c>, 2004.
- [4] NIIBE Yutaka, “Towers of Hanoi (Networking/GTK+)”, <http://www.gniibe.org/code/gtk-toh.scm>, 2004.
- [5] Paul K. Stockmeyer, “The Tower of Hanoi: A Historical Survey and Bibliography”, <http://www.cs.wm.edu/~pkstoc/biblio.ps>, 1997 (Corrected 2001).
- [6] Amit Singh, “Hanoimania”, <http://www.kernelthread.com/hanoi/>, 1994-2005.
- [7] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, “Concrete Mathematics (2nd edition)”, Addison-Wesley, 1994.
- [8] Shiro Kawai, “Gauche – A scheme implementation”, <http://www.shiro.dreamhost.com/scheme/gauche/index.html>
- [9] P. J. Hayes, “A Note on the Towers of Hanoi Problem”, *The Computer Journal* Vol. 20.3, 1977.
- [10] P. Buneman and L. Levy, “The Towers of Hanoi Problem”, *Information Processing Letters*, Vol. 10.4, 5, 1980.
- [11] Herbert Mayer, Don Perkins, “Towers of Hanoi revisited a nonrecursive surprise”, *ACM SIGPLAN Notices*, v.19 n.2, February 1984.
- [12] T. F. Hudson, Jr., “Letter to the Editor”, *SIGPLAN Notices*, v.19 n.8, August 1984.
- [13] Wm. Randolph Franklin, “A simpler iterative solution to the Towers of Hanoi problem”, *ACM SIGPLAN Notices*, v.19 n.8, August 1984.
- [14] Manfred Stadel, “Another nonrecursive algorithm for the towers of Hanoi”, *ACM SIGPLAN Notices*, v.19 n.9, September 1984.
- [15] P. J. Floriani, “Letter to the Editor”, *SIGPLAN Notices*, v.19 n.12, December 1984.
- [16] Bertrand Meyer, “A note on iterative Hanoi”, *ACM SIGPLAN Notices*, v.19 n.12, December 1984.
- [17] Graham Birtwistle, “The coroutines of Hanoi”, *ACM SIGPLAN Notices*, v.20 n.1, January 1985.
- [18] Bleicke Eggers, “The towers of Hanoi: yet another nonrecursive solution”, *ACM SIGPLAN Notices*, v.20 n.9, August 1985.