じゃんけん大会

丸山 博

1 はじめに

筆者は 1 年生を対象として開講されている情報系実習科目を担当している。1 年生前期、後期を通じて Ubuntu 上で C 言語の一通りの内容を網羅したプログラミング実習を行っており、ループや条件分岐などの項目毎に課題を課して、学生にプログラムを作成、提出をさせている。

1 年生後期の終わり頃には実習のテーマとしては総合的な内容を扱って、プログラミング実習の総まとめ的な課題を毎年課しているが、学生が興味を持って取り組んでもらう目的で、対戦型のじゃんけんプログラムを書いてもらうことにした。但し、割と限られた期間でプログラムを作ってもらうには負担が大きいので、次に出す手を考えるだけのプログラムを作成する課題とした。以下では、課題の内容や課題を行う上で必要になるプログラム群について説明していく。

2 課題概要

課題の目的はじゃんけんにおいてグー・チョキ・パーのいずれの手を出すのかを考えるアルゴリズムをプログラム化することであり、実際に対戦するために必要なプログラムはあらかじめ与えて戦略をじっくり練ってもらうことにした。また、課題の提出後に実習の履修者全員による総当たりでじゃんけんの対戦を行って、どの学生が一番優れているプログラム（アルゴリズム）を作ったのかを披露する「じゃんけん大会」を行ってプログラム作成意欲を高めることにした。

2.1 プログラム概要

じゃんけん大会を行うためのプログラムはサーバ・クライアントモデルでじゃんけんの対戦が行われる (審判役のサーバ 1 つにじゃんけんの対戦をするクライアントが 2 つ)。クライアントはサーバに接続後、サーバから相手が出した手（1 番初めは無効値）を受け取り、自分が出す手をサーバに送る。サーバは 2 つのクライアントからじゃんけんの手のデータを受け取るとそれぞれ相手側にデータを送る。このデータのやりとりを指定回数繰り返す。

2.2 じゃんけん大会ルール

1 対 1 のじゃんけんの対戦で、グー・チョキ・パーいずれかの手を出し合うことを 1000 回繰り返す。1000 回の対戦で勝った回数の多い方に 3 点、少ない方に 0 点、勝ちが同数なら双方 1 点とする勝点方式にして、じゃんけん大会で総勝点数を競ってもらうことにした。但し、クライアントからサーバにグー・チョキ・パー以外の無効データ (データの詳細は後述) を送信した場合は負け (お互いに無効データならあいこ (引き分け)) とし、また、サーバからクライアントに対戦相手の手のデータが送信されてから 1 秒以内に次の手のデータがサーバに届かなかったら、届かなかった方のクライアントはその時点で勝点 0 点扱いとして 1000 回の対戦を待たずに終了するとした。最終的に勝点で順位づけをし、勝点が並んだら総勝数で順位づけすることにした。

2.3 配布プログラム

表 1 は、この課題のために用意したプログラム群である。学生にはこれらのプログラムを配布し、じゃんけんの手を考える関数を含む myjanken.c のみを編集してクライアントの実行ファイルを作成してもらう。じゃんけんの手を考える部分以外の通信部分等は改変を防止するためにバイナリ形式で配布することにした。

表 1: 配布したプログラム群

Makefile	コンパイル情報
client.o	クライアント側のオブジェクトファイル
janken.o	サーバとクライアント共通のオブジェクトファイル
myjanken.c	次の手を考える関数を含むソースファイル
server	サーバの実行ファイル

3 プログラム詳細

3.1 送受信データ

サーバ・クライアント間でやりとりするデータは文字データである。クライアントからサーバへはクライアントの名前やじゃんけんの手が、サーバからクライアントへは相手の手のデータがそれぞれ送られる。グー・チョキ・パーはそれぞれ「G」「C」「P」の 1 バイトの文字として流れ、サーバ、クライアントは送られてきた文字データを 0〜2 の数値に変換し処理する。

3.2 サーバプログラム

サーバプログラムはじゃんけん大会においては審判の役割を果たす。今回のじゃんけん大会を開催するにあたり、以下の主な機能が必要となる。

1. 2 つのクライアントからのみ接続要請を受け付ける。
2. 規定時間内 (1 秒以内) にクライアントから返信があるかをチェックする。
3. 双方のクライアントからデータが送られてきたら、じゃんけんのジャッジをして勝敗を記録するとともに、送られてきたデータを対戦相手側に送信する。
4. 規定回数 (1000 回) の対戦が行われたら、勝点を計算する。
5. 勝敗を表示またはファイルに記録し、クライアントを切断する。

1. において、socket()、bind()、listen() でクライアントからの接続を受け付けた後、epoll_wait() 等の関数群で接続状況を監視し、2 つのクライアントからの接続を受け付けた後は新規接続を受け付けないようにして、3 人以上でのじゃんけんとならないようにした。

2. における時間設定は長時間の思考時間によってじゃんけん大会が長時間にならないようにするための措置である。この時間を監視するために、プログラムではサーバが各クライアントに対戦相手の手を送信した時刻と次の手を受信したときの時刻を計測し、そこから時間の超過がないかをチェックしている。

3. において、クライアントから 1 バイトのデータが送られるので、これを 0 から 2 の数値に換算して引き算によりじゃんけんの勝敗を求めることができる。最後に勝点の計算をするため、配列に各対戦毎の勝敗を保存している。

4. の規定回数のサーバ・クライアント間のデータ送受信が行われたら、クライアント側から接続を切断する。これによりサーバが 1000 回の対戦を終えたと判断して最終の勝敗の計算を行うようにしている。これは、対戦中にクライアントがバグ等により異常終了することが予想されるので、サーバ側では対戦中か否かはクライアントの接続によって把握することにしている。

5. においてじゃんけん大会で総対戦の勝点計算をするために、サーバは対戦毎の結果を表示またはファイル出力をするのみとして、勝点集計用のプログラムで改めて全体の勝点を計算することにしている。これにより、じゃんけん対戦と勝点計算を分離することが可能になり、対戦だけを先に済ませ、後でまとめて対戦成績処理をすることができる。また、対戦結果ファイルは CSV 形式としているために、集計用プログラムに限らず表計算ソフトで読み込むことも可能で各種の計算等が可能なようにしている。

3.3 クライアントプログラム

クライアントプログラムはサーバへの接続処理の後、規定回数回データの送信、受信を繰り返してから接続を遮断して一連の動作を終えるといった比較的シンプルな構造になっている。

接続処理部では、ユーザから指定された任意の名前をサーバに送信してクライアントのじゃんけん対戦のエントリとしている。この名前を対戦終了後の結果出力に使用している。

じゃんけんのメイン部分はサーバから送られてきたじゃんけんの手のデータを数値に変換して学生に作成してもらう関数を呼び出す。この関数からの返り値を再びじゃんけんの手のデータに逆変換してサーバに送信する。そのため、関数からの返り値が異常値である場合は、無効データ (上記「G」「C」「P」でなく「X」) をサーバに送信する。

学生に作成してもらう関数は整数値を 1 つ受け取り整数値を返す関数として作成してもらい、配布時にはサンプルとして中身がない状態である。実習ではこの関数のみを作成してクライアントプログラムを作成することにしている。

4 対戦結果集計

実習で学生にじゃんけんのアルゴリズムを考えてもらう段階では、サーバは単に審判役としてクライアント間のデータの橋渡しをするだけでよかった。しかし、じゃんけん大会としてじゃんけん対戦を複数回連続して行えるようにするために、サーバに規定回数のじゃんけん対戦を終える毎に次の対戦が出来るようクライアントの接続を改めて受け付けると共にじゃんけん対戦の結果をファイルに出力して、後で対戦結果を集計できるようにした。

規定回数のじゃんけん対戦が終わると、サーバは 1 対戦につき 1 行で結果を出力する。以下 (図 1) は対戦結果の例である。左からクライアント 1 の名前、クライアント 1 の勝点、クライアント 1 の勝数、クライアント 2 の名前、クライアント 2 の勝点、クライアント 2 の勝数、あいこの数となっている。この例では、グー・チョキ・パー・グー・チョキ・パー・…を繰り返すクライアント「guchokipa」がチョキしか出さないクライアント「choki」に 334 勝 333 敗 333 引き分け (あいこ) の成績で勝点 3 を取ったことを表している。

```
guchokipa,3,334,choki,0,333,333
```

図 1 対戦結果の出力例

じゃんけん大会において 80 人の総当たりの対戦を行うと全部で 3160 対戦となり、対戦を終えると 3160 行の結果ファイルが出来上がることになる。これを、集計するために集計用のプログラムを作成し、結果を分かりやすく見られるようにした (図 2)。これにより、じゃんけん大会を行った後すぐに結果を公表することができるので、勝利者インタビューと称して学生にアルゴリズムを解説してもらう等のプレゼンをすることが可能になっている。

```
id=[cli201], kachiten=[ 115], total_win=[26337], total_lose=[9320], total_draw=[9343]
id=[cli224], kachiten=[ 114], total_win=[31548], total_lose=[7354], total_draw=[6098]
id=[cli254], kachiten=[ 109], total_win=[28979], total_lose=[8301], total_draw=[7720]
id=[cli258], kachiten=[  99], total_win=[29508], total_lose=[8903], total_draw=[6589]
id=[cli267], kachiten=[  99], total_win=[28373], total_lose=[8851], total_draw=[7776]
id=[cli263], kachiten=[  97], total_win=[23266], total_lose=[11161], total_draw=[10573]
id=[cli259], kachiten=[  96], total_win=[30129], total_lose=[8492], total_draw=[6379]
id=[cli230], kachiten=[  94], total_win=[27910], total_lose=[10002], total_draw=[7088]
id=[cli209], kachiten=[  93], total_win=[26250], total_lose=[11199], total_draw=[7551]
id=[cli240], kachiten=[  93], total_win=[18134], total_lose=[13965], total_draw=[12901]
id=[cli241], kachiten=[  90], total_win=[28327], total_lose=[8763], total_draw=[7910]
id=[cli212], kachiten=[  90], total_win=[21685], total_lose=[9040], total_draw=[14275]
id=[cli210], kachiten=[  87], total_win=[26072], total_lose=[9932], total_draw=[8996]
id=[cli228], kachiten=[  85], total_win=[22707], total_lose=[11173], total_draw=[11120]
id=[cli280], kachiten=[  84], total_win=[28666], total_lose=[8612], total_draw=[7722]
id=[cli264], kachiten=[  84], total_win=[21189], total_lose=[12614], total_draw=[11197]
id=[cli207], kachiten=[  82], total_win=[18282], total_lose=[13252], total_draw=[13466]
id=[cli237], kachiten=[  82], total_win=[16464], total_lose=[13281], total_draw=[15255]
```

図 2 結果ファイルを読み込んで勝点でソートされた集計結果

図 2 は左からクライアント名 (id)、勝点 (kachiten)、総合勝ち数 (total_win)、総合負け数 (total_lose)、総合引き分け数 (total_draw) となっている。クライアント cli201 が勝点 115 点で 1 位となっていることを示す。結果をよく見ると、2 位となったクライアント cli224 は総合 31548 勝しているにもかかわらず引き分けの数が少ないので、勝点の取りこぼしによってクライアント cli201 に及ばなかったことが分かる。

5 おわりに

プログラミング実習において、サーバ・クライアントモデルによるじゃんけんプログラムを配布し、アルゴリズム部分のみを作成してプログラムを完成させるという課題を提示して実習を行った。サーバ・クライアント間のデータ通信部分はこの実習の範囲外として予め作成されたプログラムを使用してもらうとともに、学生には単純かつ時間をかければいくらでも完成度をあげられる課題となっていて、課題に取り組む上で興味深い仕掛けを作ることが出来たのではないかと考えている。

参考

1. Linux 関連のマニュアルページの日本語ページ
   https://linuxjm.osdn.jp/
2. [C 言語] TCP / IP でパケットの送受信を行うサンプル
   http://hensa40.cutegirl.jp/archives/932
3. Geek なページ
   https://www.geekpage.jp/programming/linux-network/book/08/8-5.php
   https://www.geekpage.jp/programming/linux-network/book/10/10-3.php