

# プログラミング言語実験

## 第3回

### コンピュータ大貧民

— コンピュータ大貧民クライアント —

2019 年 04 月 22 日、23 日

## 1 配布ファイルの構成

### 1.1 大貧民サーバと大貧民標準クライアント

/usr/local/class/daihinmin/より配布した「tndhm\_devkit\_c-20180826.tar.gz」ファイルは、大貧民サーバと大貧民標準クライアントが同梱されている。「tndhm\_devkit\_c-20180826.tar.gz」を展開したディレクトリの構成は以下のようになっている。

```
└─ tndhm_devkit_c-20180826
   └─ client
   └─ man
   └─ server
       └─ src
```

図 1: ディレクトリ構成

「client」ディレクトリには大貧民標準クライアントプログラム、「server」ディレクトリには大貧民サーバプログラム、「man」ディレクトリにはマニュアルが格納されている。この授業では上記のプログラムを改変する必要はない。この授業では、大貧民クライアントプログラムの実装を行うが、実装は以降で説明する「大貧民教育用クライアント」を対象に行う。

### 1.2 大貧民教育用クライアント

#### 1.2.1 大貧民教育用クライアントのコンパイル・実行

/usr/local/class/daihinmin/に格納されている「tndhmc-0.03.tar.gz」ファイルは、大貧民教育用クライアントのファイルであり、この授業で開発する対象のプログラムである。

まず、大貧民教育用クライアントのファイルを各自の領域にコピーして展開し、コンパイルを行う。続いて、大貧民教育用クライアントを実行する。

1. 画面左側に表示されている「端末」のアイコンをクリックし、ターミナルを起動する。
2. コンピュータ大貧民プログラムを各自のホームディレクトリ下の階層にコピーする。

```
cp /usr/local/class/daihinmin/tndhmc-0.03.tar.gz ~/uecda/
```

3. コピーしたファイルを展開する。

```
cd ~/uecda
tar xzvf tndhmc-0.03.tar.gz
```

4. 大貧民教育用クライアントのコンパイル (configure、make) を行う。

```
cd ~/uecda/tndhmc-0.03
./configure
make
```

なお、C のソースプログラムを修正し、再度コンパイルを行うときは、「make clean」を実行して実行ファイルを削除した後で、「make」を実行する。

5. (大貧民サーバを実行(「コンピュータ大貧民とは」の「3.1.1 大貧民サーバのコンパイル・実行」手順 5 参照) 後、) 大貧民教育用クライアントを実行する。

```
cd ~/uecda/tndhmc-0.03/src
./client -p <ポート番号> &
```

ポート番号は、大貧民サーバ起動時に指定したポート番号を入力する。

### 1.2.2 ソースコードの概要

大貧民教育用クライアントは、教育用途に合わせて、大貧民標準クライアントからサブセットを作成したものである。最低限の関数のみ実装し、大域変数を廃止してある。ファイル構成は以下のとおりである。

- client.c  
全体の流れを制御
- select\_cards.c  
提出カードを選択

- common.c  
共通関数
- daihinmin.c  
基本関数
- connection.c  
通信関係

次に、この授業で主に扱う common.c と select\_cards.c について説明する。  
common.c は基本的な配列操作を行う関数である。主に以下の操作を行う。

- 指定より弱いカードの削除。指定したスート<sup>1</sup>の削除
- and、or、copy、diff、not、count\_cards
- テーブルのコピー、テーブルの初期化

select\_cards.c は基本的なカード提出アルゴリズムであり、主に以下の操作を行う。

- 通常時に単騎<sup>2</sup>でのみ提出
- 革命<sup>3</sup>時や縛り<sup>4</sup>、ペアや階段<sup>5</sup>では何もしない

[バグ情報] 上記のファイル common.c 中にバグがあります。154 行目の

```
if ( (field_status->is_rev==0) || (field_status->is_rev==1) ) {
```

は、

```
if (field_status->is_rev==0) {
```

に修正しておいてください。

## 2 大貧民におけるカードの表現

UECda において、場に出ているカードや手持ちのカードは、 $8 \times 15$  の配列として与えられる。概要を表 1 に示す。UECda におけるクライアントプログラムの作成を行うには、“配列を読み取り、どのカードを出すかを決定し、やはり配列に値をセットして送り返す”プログラムを書くことになる。

たとえば、手持ちのカードがダイヤの 3、4、ハートの 9 と 11 の場合は、表 2 のようになる（カードがあるなら 1、ないなら 0）。

なお、Joker は「2」で表される。表 3 がその例である。

場にハートの 3、4、Joker、ハートの 6 の階段が出された場合は、表 4 のように表される。

<sup>1</sup>トランプのマーク（スペード、ハート、ダイヤ、クローバー）の意味。

<sup>2</sup>カード 1 枚のみで出すこと。

<sup>3</sup>革命が発生すると、カードの強弱が逆転する。革命が発生する条件は以下の通り：

( 1 ) 同じ数字のカードが 4 枚同時に出了れたとき  
( 2 ) 同じ数字のカード 3 枚と、ジョーカーが同時に出了れたとき  
( 3 ) 同じ数字のカード 4 枚とジョーカーが同時に出了れたとき  
( 4 ) 5 枚以上のカードが階段として同時に出了れたとき

<sup>4</sup>場に出ているカードと同じスートのカードしか出すことができない状態。

<sup>5</sup>同じスートの、数字の連続したカードが 3 枚以上ある状態。

表 1: コンピュータ大貧民におけるカード表現

	0	1	2	3	4	5	6	7	8	9	10	J	Q	K	A	2
0		スペード、左から 3,4,5,...,Q,K,A,2														
1		ハート														
2		ダイヤ														
3		クローバー														
4																
5																
6																
7																

表 2: 手持ちカードの表現例 1

	0	1	2	3	4	5	6	7	8	9	10	J	Q	K	A	2
0																
1								1		1						
2		1	1													
3																
4																
5																
6																
7																

表 3: 手持ちカードの表現例 2 (ジョーカー)

	0	1	2	3	4	5	6	7	8	9	10	J	Q	K	A	2
0																
1			1	1	1				1	1		1				
2							1									
3							1			1						
4		2														
5																
6																
7																

表 4: 場のカードの表現例 (階段)

		3	4	5	6	7	8	9	10	J	Q	K	A	2	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0															
1		1	1	2	1										
2															
3															
4															
5															
6															
7															

### 3 構造体 state と変数 field\_status

構造体 state と変数 field\_status の概要を以下に示す。

- 場の状況等を保存するための構造体と変数
- get\_field\_state\_from\_own\_cards と get\_field\_state\_from\_field\_cards で更新 ( client.c にすでに書いてある )
- 構造体のメンバ ( すべて整数型 )
  - ord, suit[5], quantity, is\_sequence  
( is\_sequence は場のカード情報を表す )
  - is\_rev, is\_lock, is\_no\_card  
( is\_no\_card は場の状況を表す )
  - player\_quantity[5], player\_rank[5], seat[5];  
( player\_rank[5] はプレイヤーの状況を表す )
  - have\_joker  
( 自分が Joker を持っているかどうかを表す )