

# main program

## 目次

### main program とは

main program は設定以下から exit; までを指し、動作の分岐点として SOSII の土台となる部分です

サブルーチンではないが、改造によって新たな機能を追加する際によく修正が必要になる部分でもあります。

ここでは、スクリプトのはじまりから exit; までを解説します。

### ソースコードの解説

```
#!/usr/local/bin/perl
#   サーバに合わせて設定
```

CGI を動かすためのおまじない。

Perl という CGI によく使われるプログラム言語へのパスで、これによって perl のある場所を指定しています。

サーバによって perl が置かれている場所が違うので、サーバに合わせて書き換えなければいけません。

多くの場合サーバの FAQ や仕様の書かれたページに perl へのパスが載っているので調べてみましょう。

```
# #####
# Script of Saga II
$ver = '1.10';
# Copyright (C) 2001-2003 Missing Link.
# :
# #####
```

\$ver にバージョン情報を代入。

```
require 'jcode.pl';      # jcode.pl のパス 644
```

require 関数により jcode.pl を読み込みます。

```
# ----- FILE PATH
$cgiurl = './sos2.cgi';   # sos.cgi のパス 755(705)
$sosdat = './sos.dat';   # sos.dat のパス 666(606)
$jobdat = './job.dat';   # job.dat のパス 666(606)
$itmdat = './itm.dat';  # itm.dat のパス 666(606)
:
# ----- METHOD
$method = 'POST';       # アクションメソッド (POST Only)
```

設定をひとまとめにした部分。

改造を行ったさい新しい設定が必要になった場合は、ここに追加すると都合がいい。

改造サイトで「設定に追加してください」などの指定があった場合はこの部分を指します。  
また、マシマロ分割・統合版のファイル名から、init.ini と呼ばれることもあり、  
「init.ini に追加」などの指定があった場合もこの部分を指していると思ってまず間違いありません。

```
# Main Program #
```

ここからメインプログラムと呼ばれる部分です。

```
&decode;
```

sub decode を呼び出します。

ここでフォームから送られてきた情報にデコード処理を行い、  
\$Fm{id} などの連想配列 %Fm に代入しています。

```
if ($Fm{id} = /#\W/) { &error('不正な入力です') }  
if ($Fm{pd} = /#\W/) { &error('不正な入力です') }  
if ($Fm{pd2} = /#\W/) { &error('不正な入力です') }
```

- \$Fm{id} (フォームから送られてきた、プレイヤーの ID)
- \$Fm{pd} (フォームから送られてきた、パートナーの ID)
- \$Fm{pd2}(テキストフォームに直接入力されたパートナー ID)

の 3 つの変数の中に \W (半角英数字 [a-zA-Z0-9\_] 以外の文字) が入っていればエラー処理を行います。

これはフォームから送られてきた ID を利用してファイルを開くために必要な処理で、  
この処理を行わない場合、予測しない文字列が ID に含まれると  
別のファイルを開いてしまうなどの脆弱性につながる危険性があります。

```
&lock;
```

sub lock を呼び出します。

ファイルロックを実行し、ロックが解除されるまでは他のリクエストに待機させる排他処理を行います。

```
if (!$Fm{mode}) { &main_form }  
if ($Fm{mode} eq 'norm_save') { &save_game }  
if ($Fm{mode} eq 'hide_save') { &save_game }  
if ($Fm{mode} eq 'make_new') { &make_new }  
if ($Fm{mode} eq 'make_con') { &make_con }  
if ($Fm{mode} eq 'new_game') { &enter_form }  
if ($Fm{mode} eq 'con_game') { &enter_form }  
if ($Fm{mode} eq 'find_partner') { &contact_form }  
if ($Fm{mode} eq 'find_item') { &play_form }  
if ($Fm{mode} eq 'dump_before') { &play_form }  
if ($Fm{mode} eq 'dump_after') { &play_form }  
if ($Fm{mode} eq 'goods_before') { &action_form }  
if ($Fm{mode} eq 'goods_after') { &play_form }  
if ($Fm{mode} eq 'use_before') { &action_form }  
if ($Fm{mode} eq 'use_after') { &play_form }  
if ($Fm{mode} eq 'stell_before') { &action_form }  
if ($Fm{mode} eq 'stell_after') { &play_form }  
if ($Fm{mode} eq 'stell_comp') { &play_form }  
if ($Fm{mode} eq 'judge_before') { &action_form }
```

```

if ($Fm{mode} eq 'judge_after') { &play_form }
if ($Fm{mode} eq 'trade_before') { &action_form }
if ($Fm{mode} eq 'trade_after') { &contact_form }
if ($Fm{mode} eq 'sell_before') { &action_form }
if ($Fm{mode} eq 'sell_after') { &contact_form }
if ($Fm{mode} eq 'buy_before') { &action_form }
if ($Fm{mode} eq 'buy_after') { &contact_form }
if ($Fm{mode} eq 'fee_before') { &play_form }
if ($Fm{mode} eq 'fee_after') { &play_form }
if ($Fm{mode} eq 'cr') { &contact_form }
if ($Fm{mode} eq 'rv') { &contact_form }
if ($Fm{mode} eq 'com_before') { &play_form }
if ($Fm{mode} eq 'com_after') { &play_form }
if ($Fm{mode} eq 'note_before') { &play_form }
if ($Fm{mode} eq 'note_after') { &play_form }
if ($Fm{mode} eq 'poem_before') { &play_form }
if ($Fm{mode} eq 'poem_after') { &play_form }
if ($Fm{mode} eq 'name_before') { &play_form }
if ($Fm{mode} eq 'name_after') { &play_form }
if ($Fm{mode} eq 'words_before') { &contact_form }
if ($Fm{mode} eq 'words_after') { &contact_form }
if ($Fm{mode} eq 'goodbye') { &play_form }
if ($Fm{mode} eq 'list') { &list_order }
if ($Fm{mode} eq 'list') { &player_list }
if ($Fm{mode} eq 'sale') { &sale_order }
if ($Fm{mode} eq 'my_record') { &my_record }
if ($Fm{mode} eq 'reward') { &play_form }
if ($Fm{mode} eq 'fight') { &contact_form }
if ($Fm{mode} eq 'del_before') { &play_form }
if ($Fm{mode} eq 'del_after') { &main_form }
if ($Fm{mode} eq 'back_up') { &play_form }
if ($Fm{mode} eq 'levy') { &levy }

```

\$Fm{mode} の内容によってプログラムを分岐しています。

\$Fm{mode} がなければ sub main\_form へ、  
あれば

- sub save\_game
- sub make\_new
- sub make\_con
- sub main\_form
- sub enter\_form
- sub play\_form
- sub contact\_form
- sub action\_form
- sub list\_order
- sub player\_list
- sub sale\_order
- sub my\_record
- sub levy

のうち、対応するサブルーチンへ移行します。

多くの場合、プレイヤーの行動は

- sub play\_form
- sub contact\_form
- sub action\_form

のどれかを通して実行されます。

SQS2 の改造によってプレイヤーが実行するアクションを追加する場合、この部分に

```
• if ($Fm{'mode'} eq '*****_before') { &play_form }
• if ($Fm{'mode'} eq '*****_after') { &play_form }
```

上記のようなコードを必ずと言っていいほど追加することになります。

```
&unlock;
&footer;
exit;
```

sub unlock を呼び出してファイルロックを解除し  
sub footer を呼び出して著作権表示と HTML の終了を print、  
exit 関数でプログラムを終了します。

## キーワード解説

### require EXPR

EXPR を Perl スクリプトファイルとして呼び出し、実行します。  
実行する前に指定されたファイルが既に読み込まれていないか確認します。

実行したファイルは %INC に

```
• $INC{$filename} = $realfilename;
```

形式で記録され、一度読み込んだファイルは二度呼び出されることはありません。

require は実行結果の戻り値を必要とするので、読み込むファイルの最後に 1; と書いておく必要があります。これは

```
• return 1;
```

と同等です。

### &xxxxx(ARGUMENT)

sub xxxxx を呼び出します。引数として ARGUMENT を渡すことができます。  
ARGUMENT は特殊変数 @\_ という配列に格納され、配列としてサブルーチンに渡されます。

### exit EXPR

プログラムの実行を終了してステータスコードとして EXPR の値を返します。  
EXPR が未定義の場合 0 を返します。

## 関連項目

- [コラム](#)
- [sub decode](#)
- [sub lock](#)
- [sub unlock](#)

このページのコメント