

[配布先]	<b>統数研殿向け 物理乱数生成・配送システム 機能仕様</b>	[文書番号]
		2005/12/21
		日本テクノラボ株式会社 開発部

### 改版履歴

日付	Ver.	変更内容	担当
05/06/19	0.1	入札条件を満たすために必要となる概略仕様の初版を作成	細谷
05/06/23	0.2	複数ノード、クライアントライブラリーに関する記述を追加	細谷
05/10/01	0.3	クライアントライブラリーAPI の仕様を追記	細谷
05/10/03	0.4	誤記を修正	細谷
05/12/09	0.5	エラーコード、関連ファイルに関する記述を追加、誤記を修正	黄
05/12/21	1.0	誤記を修正	細谷

## Table of Contents

<b>1</b>	<b>概要</b> .....	<b>3</b>
1.1	使用目的と設計方針 .....	3
1.2	実行環境 .....	4
<b>2</b>	<b>実装仕様</b> .....	<b>5</b>
2.1	ソフトウェアモジュール構成 .....	5
2.2	クライアント用物理乱数取得 API.....	6
2.2.1	C 言語用 API .....	6
2.2.2	Fortran 用 API .....	7
2.3	物理乱数要求受付・配送制御スレッド.....	8
2.4	物理乱数バッファプール.....	8
2.5	物理乱数取得スレッド.....	9
2.6	ランダムマスタードライバ.....	9
2.7	コンフィグレーション .....	10
2.8	エラー発生時の処理 .....	12
2.9	エラーメッセージおよびエラーの要因.....	12
2.10	関連ファイル .....	15

## 1 概要

本書は、日本ヒューレット・パッカード(株)殿經由 統計数理研究所殿向け 計算統計学支援システムにおいて、物理乱数の生成と配送するための役割を担う、物理乱数生成・配送ソフトウェアのk機能仕様について記述しております。

### 1.1 使用目的と設計方針

物理乱数生成・配送ソフトウェアは、物理乱数サーバに内蔵された東芝製物理乱数発生装置(ランダムマスター)から物理乱数を取得し、物理乱数の使用要求者(クライアント)へ入札仕様以上の性能値による配送が実現できることを目的とし、主に以下の役割を担います。

- 物理乱数サーバ1ノード毎に搭載される、4枚の東芝製物理乱数発生装置(ランダムマスター)から物理乱数を取得でき、1ノード毎に定められた物理乱数生成性能を満たすこと。
- 物理乱数の使用要求者(クライアント)側にて、効率よく物理乱数の取得を実現するために、物理乱数サーバ-クライアント間の物理乱数転送手順を透過的に実行するクライアント用APIを実装する。このクライアント用APIについては、Linux(IA32)版、Linux(IA64)版、Win32版、AIX版、SuperUNIX版(ソースコード提供による)の各プラットフォームで実行可能なC言語用ライブラリの形式で提供する。
- 物理乱数の使用要求が物理乱数サーバ1ノードの処理能力を超える場合は、物理乱数の使用要求者(クライアント)側の判断により、複数の物理乱数サーバノードの中から任意のノードを選択することによる負荷分散が可能なこと。(当初、この負荷分散については、物理乱数サーバの上位に位置する物理乱数ゲートウェイサーバにて自動的に処理する計画があったが、コスト的な問題で物理乱数ゲートウェイサーバの提供については中止となったため、今回のシステムには含まない。)

物理乱数生成・配送ソフトウェアは、次の事項を念頭に設計し、実装するものとします。

- サーバ側ソフトウェアにおいては、物理乱数サーバに搭載された4枚の東芝製物理乱数発生装置(ランダムマスター)より効率的に物理乱数を取得できるように設計し実装する。
- 取得した物理乱数を、物理乱数の使用要求者(クライアント)へ効率よく転送できるように設計し実装する。
- クライアント側ソフトウェアにおいては、複数の物理乱数サーバノードの中から任意のノードを選択して物理乱数の取得ができるように設計し実装する。
- 物理乱数の使用要求者(クライアント)へ提供するクライアント用APIを含むC言語用ライブラリは、FORTRAN77(G77)からのサブルーチン呼び出しが可能なように設計し実装する。
- 高信頼性、高可用性を意識した設計と実装を行う。

## 1.2 実行環境

物理乱数生成・配信ソフトウェアは、次の環境において実行することを前提としております。

### サーバサイド

- 日本ヒューレット・パッカード社製 ProLiant DL585 + Red Hat Enterprise Linux 3

### クライアントサイド

- Red Hat Enterprise Linux 3 (IA32 版)
- Red Hat Enterprise Linux 3 (IA64 版)
- IBM AIX 5L
- Windows XP (Win32)
- SuperUNIX

## 2 実装仕様

本節では、物理乱数生成・配信ソフトウェアの実装仕様の概略について記述します。

### 2.1 ソフトウェアモジュール構成

物理乱数生成・配信ソフトウェアは、次の図のソフトウェアモジュールにて構成し稼動するように実装しております。

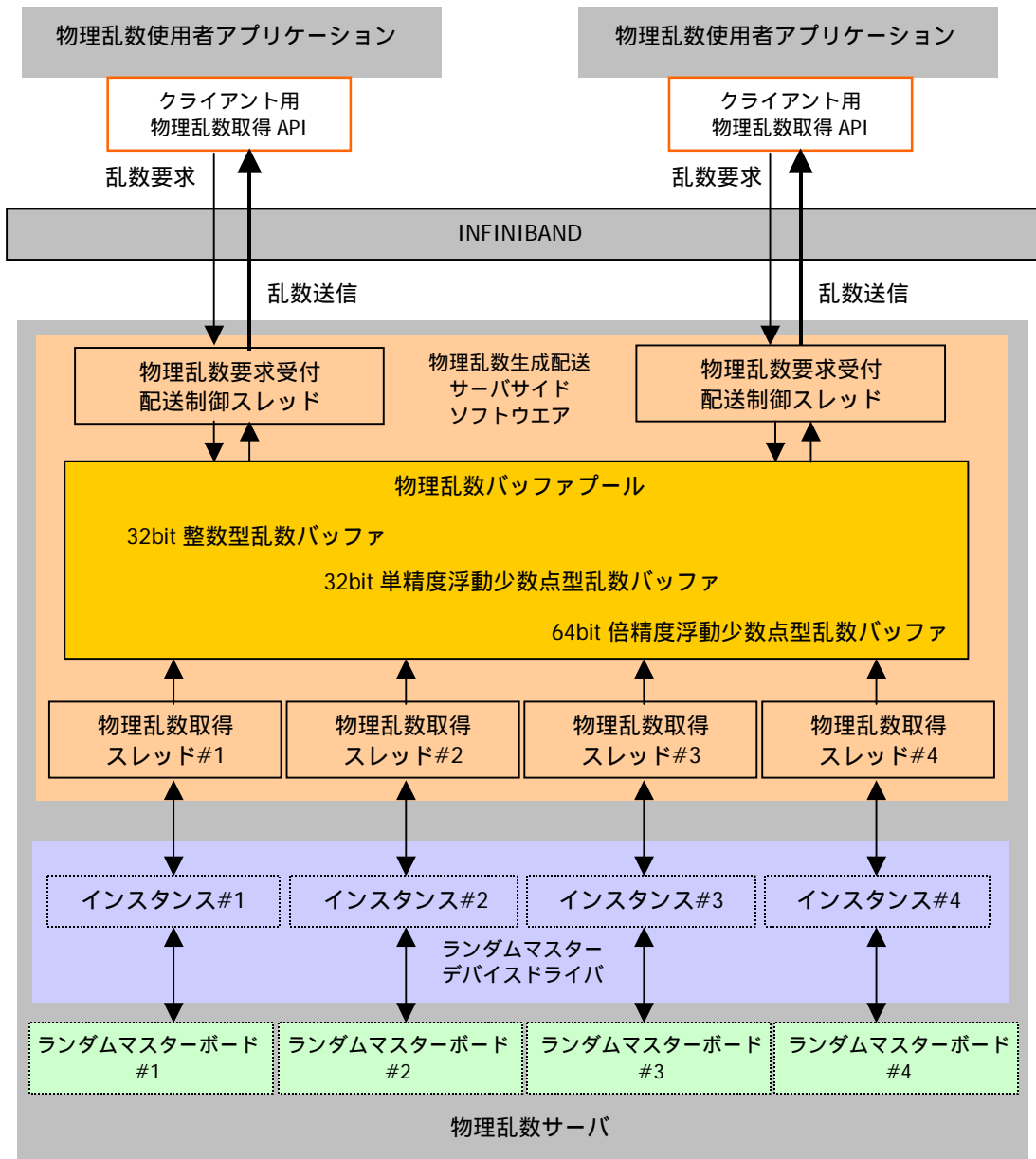


図 2.1 物理乱数生成・配信ソフトウェア構成

## 2.2 クライアント用物理乱数取得 API

クライアント用物理乱数取得 API は、物理乱数サーバ上で稼動する物理乱数生成・配送ソフトウェアとインターフェースし、TCP/IP ネットワークを介して物理乱数の転送を実現するためのソフトウェアです。物理乱数を使用するアプリケーション作成者のために、アプリケーションとリンク可能なライブラリ形式で提供いたします。クライアント用物理乱数取得 API ライブラリは、入札仕様に記載された、Linux(IA32)、Linux(IA64)、Win32、AIX、SuperUNIX にて稼動するように実装します。(ただし、SuperUNIX 版については、動作検証用機材の調達が困難であるため、クライアント用物理乱数取得 API のソースコードを提供することでの対応とさせていただきます。)クライアントライブラリについては、C 言語により実装しますが、FORTRAN77(G77)からのサブルーチン呼び出しが可能なように実装します。

### 2.2.1 C 言語用 API

API 書式	内容
<pre>#include "prandom_server.h" int RM_Init(char *)</pre>	<p>クライアント用物理乱数取得ソフトウェアを初期化します。</p> <p>第1引数には、物理乱数生成・配送ソフトウェアが稼動する物理乱数サーバの IP アドレスを指定します。</p> <p>指定された物理乱数生成・配送ソフトウェアとの接続成功によりサーバサイドとの接続識別用ハンドル(1以上の整数値)を返します。エラー発生時には、"-1"を返します。以後、他の API の利用時には、この接続識別用ハンドルが必要となります。</p>
<pre>#include "prandom_server.h" int RM_Finish(int)</pre>	<p>クライアント用物理乱数取得ソフトウェアの初期化時に確保された関連データを破棄します。</p> <p>第1引数には、接続識別用ハンドルを指定します。</p>
<pre>#include "prandom_server.h" int RM_read_int(int, int*, int)</pre>	<p>32 ビット符号付整数型の物理乱数を取得します。</p> <p>第1引数には、接続識別用ハンドルを指定します。 第2引数には、取得した物理乱数を格納するための 32 ビット符号付整数型配列のアドレスを指定します。 第3引数には、取得する 32 ビット符号付整数型物理乱数の個数を指定します。</p> <p>返り値は、取得した物理乱数の個数を返します。エラー発生時には、"-1"を返します。</p>
<pre>#include "prandom_server.h" int RM_read_float(int, float*, int)</pre>	<p>32 ビット単精度浮動小数点型の物理乱数を取得します。</p> <p>第1引数には、接続識別用ハンドルを指定します。 第2引数には、取得した物理乱数を格納するための 32 ビット単精度浮動小数点型配列のアドレスを指定します。 第3引数には、取得する 32 ビット単精度浮動小数点型物理乱数の個数を指定します。</p> <p>返り値は、取得した物理乱数の個数を返します。エラー発生時には、"-1"を返します。</p>

<pre>#include "prandom_server.h" int RM_read_double(int, double*, int)</pre>	<p>64 ビット倍精度浮動小数点型の物理乱数を取得します。</p> <p>第1引数には、接続識別用ハンドルを指定します。 第2引数には、取得した物理乱数を格納するための64ビット倍精度浮動小数点型配列のアドレスを指定します。 第3引数には、取得する64ビット倍精度浮動小数点型物理乱数の個数を指定します。</p> <p>返り値は、取得した物理乱数の個数を返します。エラー発生時には、"-1"を返します。</p>
--	--

[注意点]

物理乱数取得関数は"RM\_Init" と"RM\_Finish" の間で使用してください。これはプログラムの最初と最後に一度だけ行います。

## 2.2.2 Fortran 用 API

API 書式	内容
<pre>rmnit(integer, character)</pre>	<p>クライアント用物理乱数取得ソフトウェアを初期化します。</p> <p>第1引数は、サブルーチンコールのチェックサムです。指定された物理乱数生成・配信ソフトウェアとの接続成功によりサーバサイドとの接続識別用ハンドル(1以上の整数値)が設定されます。エラー発生時は、"-1"が設定されず。以後、他のAPIの利用時には、この接続識別用ハンドルが必要となります。</p> <p>第2引数には、物理乱数生成・配信ソフトウェアが稼動する物理乱数サーバのIPアドレスを指定します。</p>
<pre>rmffinish(integer, integer)</pre>	<p>クライアント用物理乱数取得ソフトウェアの初期化時に確保された関連データを破棄します。</p> <p>第1引数は、サブルーチンコールのチェックサムです。(成功時は0、失敗時は1を返します。) 第2引数には、接続識別用ハンドルを指定します。</p>
<pre>rmri(integer, integer, integer, integer)</pre>	<p>32ビット符号付整数型の物理乱数を取得します。</p> <p>第1引数は、サブルーチンコールのチェックサムです。(成功時は、取得した物理乱数の個数を返します。エラー発生時には、"-1"を返します。) 第2引数には、接続識別用ハンドルを指定します。 第3引数には、取得した物理乱数を格納するための32ビット符号付整数型配列のアドレスを指定します。 第4引数には、取得する32ビット符号付整数型物理乱数の個数を指定します。</p>
<pre>rmrf(integer, integer, real*4, integer)</pre>	<p>32ビット単精度浮動小数点型の物理乱数を取得します。</p> <p>第1引数は、サブルーチンコールのチェックサムです。(成功時は、取得した物理乱数の個数を返します。エラー発生</p>

	<p>時には、"-1"を返します。)</p> <p>第2引数には、接続識別用ハンドルを指定します。</p> <p>第3引数には、取得した物理乱数を格納するための32ビット単精度浮動小数点型配列のアドレスを指定します。</p> <p>第4引数には、取得する32ビット単精度浮動小数点型物理乱数の個数を指定します。</p>
<p>rmrd(integer, integer, real*8, integer)</p>	<p>64ビット倍精度浮動小数点型の物理乱数を取得します。</p> <p>第1引数は、サブルーチンコールのチェックサムです。(成功時は、取得した物理乱数の個数を返します。エラー発生時には、"-1"を返します。)</p> <p>第2引数には、接続識別用ハンドルを指定します。</p> <p>第3引数には、取得した物理乱数を格納するための64ビット倍精度浮動小数点型配列のアドレスを指定します。</p> <p>第4引数には、取得する64ビット倍精度浮動小数点型物理乱数の個数を指定します。</p> <p>返り値は、取得した物理乱数の個数を返します。エラー発生時には、"-1"を返します。</p>

[注意点]

物理乱数取得関数は、"rminit" と"rmfinish" の間で使用してください。これはプログラムの最初と最後に一度だけ行います。

## 2.3 物理乱数要求受付・配送制御スレッド

物理乱数要求受付配送制御スレッドは、物理乱数使用者(クライアント)とのインターフェースを担うモジュールで、要求に合致する型式の物理乱数を物理乱数バッファプールから取得し、物理乱数使用者(クライアント)へ配送するためのソフトウェアです。

表 2.3 配送乱数形式一覧

型式	乱数値の範囲
32ビット符号付整数型	区間[-2 <sup>31</sup> , 2 <sup>31</sup> )
32ビット単精度浮動小数点型	区間[0,1)
64ビット倍精度浮動小数点型	区間[0,1)

## 2.4 物理乱数バッファプール

物理乱数バッファプールは、物理乱数取得スレッドがランダムマスタードライバを経由して物理乱数発生装置から取得した物理乱数を蓄積するためのバッファです。物理乱数バッファプールを介して物理乱数の受け渡しをするため、物理乱数要求受付配送制御スレッドと物理乱数取得スレッドとは、非同期的に動作します。

物理乱数バッファプールは、32ビット整数型乱数を格納するためのバッファ、32ビット単精度浮動小数点型乱数を格納するためのバッファ、64ビット倍精度浮動小数点型乱数を格納するためのバッファに区分けされており、それぞれのバッファのサイズは任意な数値に設定可能ですので、使用頻度の高い型式の乱数用に多くのバッファを割り当てることにより、物理乱数配送の効率化を図れます。



## 2.5 物理乱数取得スレッド

物理乱数取得スレッドは、物理乱数バッファプールの水位(蓄積された物理乱数数量)をトリガーとして、ランダムマスタードライバを経由して物理乱数発生装置から物理乱数を取得するためのソフトウェアです。

## 2.6 ランダムマスタードライバ

ランダムマスタードライバは、物理乱数取得モジュールから通知された型式の物理乱数を物理乱数発生装置に発生させ、物理乱数発生装置が生成した物理乱数を DMA 転送にてホストとなるサーバのメインメモリーへ転送するためのソフトウェアです。

## 2.7 コンフィグレーション

物理乱数生成・配信ソフトウェアの実行時に必要となるコンフィグレーションパラメータは、本ソフトウェアを実行するマシンの次のパスに置かれた設定ファイルから取得します。

表 2.7 物理乱数生成・配信ソフトウェア設定一覧

格納場所	設定ファイル		
格納先パス名	/opt/NTLPrandom/etc/prandom_server.conf		
設定項目	[セクション]/キー	パラメータとその意味	備考
物理乱数発生装置数	NumOfPrandom=	制御対象の物理乱数発生装置(ランダムマスター)の数を整数にて設定します。未設定時のデフォルトは、4(枚)となっております。	
32ビット整数型物理乱数バッファプールサイズ	NumOfDwordBuffer=	物理乱数生成・配信ソフトウェアが 32 ビット整数型乱数用に確保する物理乱数バッファの個数を整数にて設定します。未設定時のデフォルトは、128(個)となっております。 1 個あたりの物理乱数バッファのサイズは、物理乱数発生装置(ランダムマスター)間との DMA 転送が最適に実行できる 128KB(131072 バイト)固定となっております。物理乱数バッファプール全体のサイズは、(SizeOfDwordBuffer=で指定した数値 + SizeOfFloatBuffer=で指定した数値 + SizeOfDoubleBuffer=で指定した数値) x 128KB となります。	
32ビット単精度浮動小数点型物理乱数バッファプールサイズ	NumOfFloatBuffer=	物理乱数生成・配信ソフトウェアが 32 ビット単精度浮動小数点型乱数用に確保する物理乱数バッファの個数を整数にて設定します。未設定時のデフォルトは、128(個)となっております。	
64ビット倍精度浮動小数点型物理乱数バッファプールサイズ	NumOfDoubleBuffer=	物理乱数生成・配信ソフトウェアが 64 ビット倍精度浮動小数点型確保する物理乱数バッファの個数を整数にて設定します。未設定時のデフォルトは、128(個)となっております。	
接続クライアント数	MaxNumOfClient=	この物理乱数生成・配信ソフトウェアに接続可能なクライアント数の最大値を整数にて設定します。未設定時のデフォルトは、64(クライアント)となっております。	

ポート番号	ClientPortNo=	クライアント用物理乱数取得 API からの TCP/IP による接続を待ち受けるポート番号を整数にて設定します。未設定時のデフォルトは、9999 となっておりますので、統計数理研究所殿の既存環境において他のアプリケーションで使用しているポート番号と重複しない番号を設定してください。	

## 2.8 エラー発生時の処理

物理乱数生成・配信ソフトウェアは、ソフトウェア内部で検知したエラーについては、そのエラーを示す一意のエラーコードとエラーメッセージをシステムログに記録します。エラー発生時の通知内容および、各エラーコードとその意味については、後述する「エラーコード一覧」を参照してください。

表 2.8 システムログ設定一覧

設定項目	内容
識別名(ident)	prandom_server
分類(facility)	LOG_DAEMON
出力先ログファイル名	RHEL3 のデフォルト設定では、/var/log/messages に記録されます。

## 2.9 エラーメッセージおよびエラーの要因

エラーコード (サーバ)	エラーメッセージおよびエラー要因	備考
1	open() error on configuration file 名 設定ファイルが開けません。所定の場所に設定ファイルが存在することを確認してください。	
2	fork() error[process id] デーモン化のためのクローンプロセスの生成ができません。システム環境に異常が無いことを確認してください。	
3	setsid() error[sid] デーモン化したプロセスのセッションプロセス・グループ ID を設定することができません。システム環境に異常が無いことを確認してください。	
4	chdir() error to server home directory 本プロセスのホームディレクトリとなる /opt/NTLPrandom/bin ディレクトリへ移動することができません。所定の場所にディレクトリが存在することを確認してください。	
5	malloc() error[errno] メモリの確保に失敗しました。実行環境にメモリ枯渇となる要因が無いことを確認してください。	
6	socket() error[errno] ソケットの作成に失敗しました。実行環境に異常が無いことを確認してください。	
7	setsockopt() error[errno] ソケットオプションの設定に失敗しました。実行環境に異常が無いことを確認してください。	
8	bind() error[errno] ソケットの名前付けに失敗しました。実行環境に異常が無いことを確認してください。	
9	listen() error[errno] 作成したソケットへの接続要求待ちに失敗しました。実行環境に異常が無いことを確認してください。	
11	pumpthread: error on open(path), errno=%d path で示されたデバイスファイル名を持つランダムマスターデバイスドライバインスタンスのオープンに失敗しました。ランダムマスターハードウェアが正しくシステムに認識されていないか、ハードウェア故障の可能性がります。	

12	pumpthread: error on ioctl(%s), errno= オープンに成功したランダムマスターデバイスドライバインスタンスに対して、生成乱数型式(32 ビット整数型乱数)の指定に失敗しました。デバイスドライバの不具合か、実行環境の異常に原因がある可能性があります。	
16	pumpthread: error on read(path) rc=, errno= オープンに成功したランダムマスターデバイスドライバインスタンスからの乱数の取得に失敗しました。ハードウェア故障もしくは、デバイスドライバの不具合か、実行環境の異常に原因がある可能性があります。	
18	accept() error[errno] 作成したソケットへの接続要求受諾処理に失敗しました。実行環境に異常が無いことを確認してください。	
21	接続先 IP アドレス: recv() error (socket=%d) 接続先からのコマンドの受信に失敗しました。	
23	接続先 IP アドレス: error on send() rc=%d, errno=%d. errpnt(%d) 接続先への乱数送信に失敗しました。	
100	pthread_create() error[errno] スレッドの作成に失敗しました。実行環境に異常が無いことを確認してください。	
102	pthread_detach error, errno スレッドの切り離しに失敗しました。実行環境に異常が無いことを確認してください。	
104	pthread_mutex_lock() error[errno] 排他制御用 API (pthread_mutex_lock) の呼び出しに失敗しました。実行環境に異常が無いことを確認してください。	
105	pthread_mutex_unlock() error[errno] 排他制御用 API (pthread_mutex_unlock) の呼び出しに失敗しました。実行環境に異常が無いことを確認してください。	
106	pthread_cond_broadcast() error[errno] 同期制御用 API (pthread_cond_broadcast) の呼び出しに失敗しました。実行環境に異常が無いことを確認してください。	
107	pthread_cond_signal() error[errno] 同期制御用 API (pthread_cond_signal) の呼び出しに失敗しました。実行環境に異常が無いことを確認してください。	
108	pthread_cond_wait() error[errno] 同期制御用 API (pthread_cond_wait) の呼び出しに失敗しました。実行環境に異常が無いことを確認してください。	

エラーコード linux/AIX 版 クライアントラ イブラリ	エラーメッセージおよびエラー要因	備考
1	RM_Init: socket() failed (sockfd=)	
2	RM_Init: invalid ip address (ip=)	
3	RM_Init: connect() failed (sockfd=)	
11	RM_read_int: requested number is less than 0 (num=)	
12	RM_read_int: send() failed (sockfd=)	
13	RM_read_int: recv() failed (sockfd=)	
21	RM_read_float: requested number is less than 0 (num=)	
22	RM_read_float: send() failed (sockfd=)	

23	RM_read_float: recv() failed (sockfd=)	
31	RM_read_double: requested number is less than 0 (num=)	
32	RM_read_double: send() failed (sockfd=)	
33	RM_read_double: recv() failed (sockfd=)	
91	RM_Finish: send() failed (sockfd=)	

エラーコード windows 版 クライアントラ イブラリ	エラーメッセージおよびエラー要因	備考
-8	SendMsg: send() failed (ret=)	
-9	RecvMsg: ioctlsocket() failed (ret=)	
-10	RecvMsg: recv() failed (ret=)	
-11	SocketCheck: WaitForSingleObject() failed (ret=)	
-12	SocketCheck: WSAEnumNetworkEvents():1 failed (ret=)	
-12	SocketCheck: WSAEnumNetworkEvents():2 failed (ret=)	
-106	SendMsg: SendMsg() timeout !!	
-106	RecvMsg: RecvMsg() timeout !!	
-106	SocketCheck: SocketCheck timeout !!	
1	RM_Init: socket() failed (sockfd=)	
2	RM_Init: invalid ip address (ip=)	
3	RM_Init: connect() failed (sockfd=)	
11	RM_read_int: requested number is less than 0 (num=)	
12	RM_read_int: SendMsg() failed (sockfd=)	
13	RM_read_int: RecvMsg() failed (sockfd=)	
21	RM_read_float: requested number is less than 0 (num=)	
22	RM_read_float: SendMsg() failed (sockfd=)	
23	RM_read_float: RecvMsg() failed (sockfd=)	
31	RM_read_double: requested number is less than 0 (num=)	
32	RM_read_double: SendMsg() failed (sockfd=)	
33	RM_read_double: RecvMsg() failed (sockfd=)	
81	RM_Init: atexit(WSACleanup) failed	
82	RM_Init: WSACreateEvent() failed (error=)	
83	RM_Init: WSACreateEvent() failed (error=)	
84	RM_Init: WSAEventSelect() failed (sockfd=)	
91	RM_Finish: send() failed (sockfd=)	

## 2.10 関連ファイル

物理乱数生成・配信ソフトウェアは、以下に示すファイルにより構成されます。  
下記のファイルは、本ソフトウェアを実行するローカルマシン上に配置されていることを前提とします。

サーバ関連ファイル(/opt/NTLPrandom/以下から記載)

ファイル名	内容
bin/prandom_server	サーバプログラム
etc/prandom_server.conf	サーバコンフィグレーションファイル

Linux-ibm クライアント関連ファイル

ファイル名	内容
README.txt	コンパイルの説明
prandom_server.h	Linux-ibm クライアント用 C ヘッダファイル
RM_Api.c	Linux-ibm クライアント Api 用 C ソース
RM_test3.c	Linux-ibm クライアントテスト用 C ソース
rmf77test.f	Linux-ibm クライアントテスト用 F77 ソース
prandom_client.log	クライアントログファイル(filepath は C ヘッダファイル中で定義)

windows クライアント関連ファイル

ファイル名	内容
READMEsjis.txt	コンパイルの説明
READMEeuc.txt	コンパイルの説明
prandom_server.h	windows クライアント用 C ヘッダファイル
prandom.cpp	windows クライアント Api 用 CPP ソース
prandom_test.cpp	windows クライアントテスト用 CPP ソース
prandom_client.log	クライアントログファイル(filepath は C ヘッダファイル中で定義)