

Paxosファイルシステムのツール概要書

1. PFSSの試用運用のためのTOOL概要

- ・ PFSSの試用運用のためのシェルスクリプト
- ・ PFSSClientのコマンドを使用したもの、Linuxのコマンドを使用した試用支援ツールなどで構成

2. PFSSServerコマンド仕様

- ・PFSSServer

PFSSServerの管理、モニターコマンド仕様

- ・ admin
- ・ PFSSProbe

3. PFSSClientのコマンド仕様

- ・ サンプルクライアントプログラム (PFSSClientMain.c) で提供されているコマンド

平成22年12月
平成23年4月改訂

PFSの試用運用のためのTOOL概要

	TOOLコマンド	概 要
サーバアプリケーション起動、停止TOOL		
1	server-loop-log.sh	<p><u>PFSServerが終了したら起動する処理を無限に繰り返す</u> USAGE: server-loop-log.sh n [log_size [log_blocks]] n: サーバID log_size: LOG Bufferサイズ (LOGファイルのサイズ) (Byte単位) log_blocks: 保存するLOG Fileの最大数</p> <ul style="list-style-type: none"> ・ server-loop-log.sh n : No Logging Mode. 最小限のログを起動端末へ出力 server-loop-log.sh n 0 : Logging Mode. 全てのログを端末へ出力 server-loop-log.sh n log_size log_blocks : Logging Mode. ログをファイルへ出力 <ul style="list-style-type: none"> ・ PFSServer n PFS-n [autonomic] の起動 ・ PFSServerの初回起動は「PFSServer n PFS-n」非autonomicモード ・ PFSServerがEXIT後30秒後に再起動(autonomicモード) ・ PFSServerがcore DUMP EXITの場合は、再起動せずserver-loop-log.sh終了 ・ PFSServerが再起動異常 (10秒以内でEXIT) 3回連続した場合server-loop-log.sh終了 ・ EXIT情報を/tmp/PFSServer-n-Exit.logに出力
2	server-loop-logAuto.sh	<p><u>PFSServerが終了したら起動する処理を無限に繰り返す</u> USAGE: server-loop-logAuto.sh n [logsize [logblocks]] n: サーバID logsize: LOG Bufferサイズ (LOGファイルのサイズ) (Byte単位) logblocks: 保存するLOG Fileの最大数</p> <p>※server-loop-log.shと機能的に下記以外は同じ 常に「PFSServer n PFS-n autonomic」とautonomicモードで起動する</p>
3	server-loop-log-cygwin.sh	<p><u>PFSServerが終了したら起動する処理を無限に繰り返す</u> ※ cygwin環境用のScript USAGE: server-loop-log-cygwin.sh n [logsize [logblocks]] n: サーバID log_size: LOG Bufferサイズ (LOGファイルのサイズ) (Byte単位) log_blocks: 保存するLOG Fileの最大数</p> <ul style="list-style-type: none"> ・ server-loop-log-cygwin.sh n : No Logging Mode. 最小限のログを起動端末へ出力 server-loop-log-cygwin.sh n 0 : Logging Mode. 全てのログを端末へ出力 server-loop-log-cygwin.sh n logsize log_blocks : Logging Mode. ログをファイルへ出力 <ul style="list-style-type: none"> ・ PFSServer n PFS-n [autonomic] の起動 ・ PFSServerの初回起動は「PFSServer n PFS-n」非autonomicモード ・ PFSServerがEXIT後40秒後に再起動(autonomicモード)
4	server-loop-cygwin.sh	<p><u>PFSServerが終了したら起動する処理を無限に繰り返す</u> ※ cygwin環境用のScript USAGE: server-loop-cygwin.sh n n: サーバID</p> <ul style="list-style-type: none"> ・ PFSServer n PFS-n [autonomic] の起動 ・ PFSServerの初回起動は「PFSServer n PFS-n」非autonomicモード ・ PFSServerがEXIT後40秒後に再起動(autonomicモード)
5	rm-serverkill.sh	<p><u>各サーバのPFSServer を循環的(400秒間隔)に強制終了(SIGKILL によるKILL)</u> <u>可能な限りMaster Serverを強制終了する</u> USAGE : rm-serverkill.sh 強制終了させるサーバ上のserver-kill.shをリモート起動 rsh paxos03 ~/PAXOS/Paxos/PFS/bin/server-kill.sh rsh paxos04 ~/PAXOS/Paxos/PFS/bin/server-kill.sh rsh paxos05 ~/PAXOS/Paxos/PFS/bin/server-kill.sh rsh paxos06 ~/PAXOS/Paxos/PFS/bin/server-kill.sh rsh paxos07 ~/PAXOS/Paxos/PFS/bin/server-kill.sh</p>
6	rm-server-randkill.sh	<p><u>各サーバのPFSServer をランダム(300秒間隔)に強制終了(SIGKILL によるKILL)</u> USAGE : rm-server-randkill.sh 強制終了させるサーバ上のserver-kill.shをリモート起動 rsh paxos0n ~/PAXOS/Paxos/PFS/bin/server-kill.sh (n:3-7)</p>
7	server-kill.sh	<p><u>サーバマシン上のPFSServerのPIDを調べ そのプロセスにSIGKILLを発行する</u> (サーバ側でリモート起動される) USAGE : server-kill.sh</p>
8	server-kill-cygwin.sh	<p><u>PFSServer を循環的(200秒間隔)に強制終了(SIGKILL によるKILL)</u> ※ cygwin環境用のScript USAGE : server-kill-cygwin.sh PFSServerのPIDを調べ そのプロセスにSIGKILLを発行する</p>

PFSSの試用運用のためのTOOL概要

TOOLコマンド	概 要
Clientアプリケーション試験TOOL (Clientマシン上で使用する)	
9	<p>f-loopwrite-log.sh</p> <p><u>「PFSSClient lockwriteloo sample.c w 500」を繰り返し起動する処理</u> USAGE: f-loopwrite-log.sh [log_size [log_blocks]] log_size: LOG Bufferサイズ (LOGファイルのサイズ) (Byte単位) log_blocks: 保存するLOG Fileの最大数</p> <ul style="list-style-type: none"> • f-loopwrite-log.sh : No Logging Mode. 最小限のログを端末へ出力 f-loopwrite-log.sh log_size : Logging Mode. 全てのログを端末へ出力 f-loopwrite-log.sh log_size log_blocks : Logging Mode. ログをファイルへ出力 <ul style="list-style-type: none"> • 「PFSSClient lockwriteloo sample.c w 500」 サーバへファイルsample.cをファイルw-001, ..., w-500として順に書き込む処理 ※ sample.c 16KB (OUTBOUND File) ※ (SessionOpen-PFSSLockW-PFSSOpen-PFSSWrite-PFSSUnlock-SessionClose) x500回 • 「PFSSClient lockwriteloo sample.c w 500」 x 10回の処理時間を/tmp/cost_WIに出力 • PFSSClientがエラーEXITの場合は、f-loopwrite-log.sh終了
10	<p>f-loopread-log.sh</p> <p><u>「PFSSClient lockreadloop w Tfile 500」を繰り返し起動する処理</u> USAGE: f-loopread-log.sh [log_size [log_blocks]] log_size: LOG Bufferサイズ (LOGファイルのサイズ) (Byte単位) log_blocks: 保存するLOG Fileの最大数</p> <ul style="list-style-type: none"> • f-loopread-log.sh : No Logging Mode. 最小限のログを端末へ出力 f-loopread-log.sh log_size : Logging Mode. 全てのログを端末へ出力 f-loopread-log.sh log_size log_blocks : Logging Mode. ログをファイルへ出力 <ul style="list-style-type: none"> • 「PFSSClient lockreadloop w Tfile 500」 サーバからファイルw-001, ..., w-500を順に読み込む処理 Client (paxos00)のPFSS/bin下にTfile-001, ..., Tfile-500が順に作成される ※ (SessionOpen-PFSSLockR-PFSSOpen-PFSSRead-PFSSUnlock-SessionClose) x500回 • 「PFSSClient lockreadloop w Tfile 500」 x 10回の処理時間を/tmp/cost_Rに出力
11	<p>f-2rrr-log.sh</p> <p><u>「PFSSClient lockreadloop2 w Ufile 500」を繰り返し起動する処理</u> USAGE: f-2rrr-log.sh [log_size [log_blocks]] log_size: LOG Bufferサイズ (LOGファイルのサイズ) (Byte単位) log_blocks: 保存するLOG Fileの最大数</p> <ul style="list-style-type: none"> • f-2rrr-log.sh : No Logging Mode. 最小限のログを端末へ出力 f-2rrr-log.sh log_size : Logging Mode. 全てのログを端末へ出力 f-2rrr-log.sh log_size log_blocks : Logging Mode. ログをファイルへ出力 <ul style="list-style-type: none"> • 「PFSSClient lockreadloop2 w Ufile 500」 サーバからファイルw-001, ..., w-500を順に読み込む 処理 Client (paxos00)のPFSS/bin下にUfile-001, ..., Ufile-500が順に作成 ※ SessionOpen - (PFSSLockR-PFSSOpen-PFSSRead-PFSSUnlock) x500回 - SessionClose
12	<p>f-3rrr-log.sh</p> <p><u>「PFSSClient lockread w-nnn Rfile-nnn」を繰り返し起動する処理</u> USAGE: f-3rrr-log.sh [log_size [log_blocks]] log_size: LOG Bufferサイズ (LOGファイルのサイズ) (Byte単位) log_blocks: 保存するLOG Fileの最大数</p> <ul style="list-style-type: none"> • f-3rrr-log.sh : No Logging Mode. 最小限のログを端末へ出力 f-3rrr-log.sh log_size : Logging Mode. 全てのログを端末へ出力 f-3rrr-log.sh log_size log_blocks : Logging Mode. ログをファイルへ出力 <ul style="list-style-type: none"> • 「PFSSClient lockread w-nnn Rfile-nnn」 サーバからファイルw-nnnを順に読み込む処理 Client (paxos00)のPFSS/bin下にRfile-nnnが作成 ※ SessionOpen-PFSSLockR-PFSSOpen-PFSSRead-PFSSUnlock-SessionClose
13	<p>f-eventdir-log.sh</p> <p><u>「PFSSClient eventdir2 temp 0 n」を起動しディレクトリ temp に関するDirectory Eventを受信する処理</u> USAGE: f-eventdir-log.sh n n: Event受信回数</p> <ul style="list-style-type: none"> • ログをファイルへ出力 • 「PFSSClient eventdir2 temp 0 n」を起動
14	<p>f-eventlock-log.sh</p> <p><u>「PFSSClient eventlockW LCKtemp/wfile n」を起動しロック名LCKtemp/wfile に関するEventを受信する処理</u> USAGE: f-eventlock-log.sh n n: Event受信回数</p> <ul style="list-style-type: none"> • ログをファイルへ出力 • 「PFSSClient eventlockW LCKtemp/wfile n」を起動

PFSSの試用運用のためのTOOL概要

	TOOLコマンド	概 要
15	f-loopeventgenerator.sh	指定サーバー(sid)の保持しているロック管理情報を得る
16	f-RW.sh	<u>ファイルの書き込み、読み込みを交互に繰り返す処理</u> USAGE: f-RW.sh PFSSClient lockwrite sample.c rw-nnn (nnn=1-500) PFSSClient lockread rw-nnn RWfile--nnn (nnn=1-500)
17	f-read.sh	<u>ファイルのロック読み込みを繰り返す処理</u> USAGE: f-read.sh PFSSClient lockread w-nnn Tfile-nnn (nnn=1-500) ※下記の読み込みも可能 PFSSClient read w-nnn Tfile-nnn (nnn=1-500) PFSSClient read temp/f-nnn Tfile-nnn (nnn=1-500)
18	f-looplockW.sh	<u>ファイルのロック書き込みを繰り返す処理</u> Usage : f-looplockW.sh PFSSClient lockW2 lock-w sleep_time
19	f-write.sh	<u>アウトバウンドファイルのロック書き込みを繰り返す処理</u> USAGE: f-write.sh PFSSClient lockwrite zoutbound100KB w-nnn (nnn=1-500)
20	f-ephemeral.sh	<u>エフェメラルファイルをtempディレクトリ下に書き込みを繰り返す処理</u> USAGE: f-ephemeral.sh PFSSClient ephemeral3 zephemeral temp
21	f-temp-cre.sh	<u>tempディレクトリ下にファイルの作成、ディレクトリの作成・削除を繰り返す処理</u> 処理終了時にSNAPSHOTを作成する USAGE: f-temp-cre.sh PFSSClient mkdir temp/work PFSSClient lockwrite zinbound temp/w-nnn (nnn=1-500) PFSSClient rmdir temp/work PFSSClient snapshot
22	f-temp-del.sh	<u>tempディレクトリ下のファイルを削除を繰り返す処理</u> USAGE: f-temp-del.sh PFSSClient delete temp/w-nnn (nnn=1-500)

PFSSの試用運用のためのTOOL概要

	TOOLコマンド	概 要
運用関連のTOOL		
23	rcp-initenv. sh	<u>paxos03/04/05/06/07の初期環境の構築する</u> USAGE: rcp-initenv. sh
24	rcp-paxos-bin. sh	<u>paxos03/04/05/06/07へ PFSServer, PFSClient, PFSProbeをコピー</u> USAGE: rcp-paxos-bin. sh
25	rcp-admin-bin. sh	<u>paxos03/04/05/06/07へ Paxos/paxos/bin/adminをコピー</u> <u>paxos03/04/05/06/07へ NWGadget/bin/LogFilesをコピー</u> <u>paxos03/04/05/06/07へ NWGadget/bin/LogPrintをコピー</u> USAGE: rcp-admin-bin. sh
26	rcp-tool-bin. sh	<u>paxos03/04/05/06/07へ 指定TOOLファイルをコピー (Paxos/PFS/bin 配下を前提)</u> USAGE: rcp-tool-bin. sh TOOL_File_Name
27	rm-delper. sh	<u>各マシン (サーバ、クライアント) の作成ファイルなどのクリア</u> 各サーバのdelper. shをリモート起動 USAGE: rm-delper. sh
28	delper. sh	<u>起動マシン上のPFS/bin下のPFS-* など不要ファイルの削除 (リモート起動)</u> USAGE: delper. sh
29	vm-status. sh	<u>各マシン上のFSServer、PFSClientプロセスのVmSize、Threadsの状態情報出力</u> USAGE: vm-status. sh vm-status-Single. sh dum wc-filenum. sh n をリモート起動
30	vm-status-Single. sh	<u>パラメータ数=1: (リモート起動の場合)</u> <u>起動マシン上のPFSServer または PFSClientプロセスのVmSize、Threadsの情報出力</u> <u>起動マシン上のPFS/bin/PFS-n/DATA下のファイル数情報出力</u> USAGE: vm-status-Single. sh [dummy]
31	vm-auto-status. sh	<u>各マシン上のFSServer、PFSClientプロセスのVmSize、Threadsの状態情報出力</u> USAGE: vm-auto-status. sh ・ 1700 Sec毎に実行 ・ vm-status. sh をリモート起動
32	wc-filenum. sh	<u>PFS/bin/PFS-n/DATA下のファイル数を調べる</u> USAGE: wc-filenum. sh server_ID (vm-status-Single. shからリモート起動される)
33	rm-netstat. sh	<u>各サーバ(paxos03/04/05/06/07)のtcp残存コネクション状態情報出力</u> USAGE: rm-netstat. sh
34	rm-admin-control. sh	<u>PAXOSの制御状態取得のadminコマンドのリモート起動</u> Usage: rm-admin-control. sh ...全サーバ対象 rm-admin-control. sh server-ID ...指定サーバ対象 rsh paxos03 ~/PAXOS/Paxos/paxos/bin/admin 0 control 1 rsh paxos04 ~/PAXOS/Paxos/paxos/bin/admin 1 control 1 rsh paxos05 ~/PAXOS/Paxos/paxos/bin/admin 2 control 1 rsh paxos06 ~/PAXOS/Paxos/paxos/bin/admin 3 control 1 rsh paxos07 ~/PAXOS/Paxos/paxos/bin/admin 4 control 1
35	rm-admin-paxos. sh	<u>PAXOS状態情報取得のadminコマンドのリモート起動</u> USAGE: rm-admin-paxos. sh ...全サーバ対象 rm-admin-paxos. sh server-ID ...指定サーバ対象 rsh paxos03 ~/PAXOS/Paxos/paxos/bin/admin 0 paxos 1 rsh paxos04 ~/PAXOS/Paxos/paxos/bin/admin 1 paxos 1 rsh paxos05 ~/PAXOS/Paxos/paxos/bin/admin 2 paxos 1 rsh paxos06 ~/PAXOS/Paxos/paxos/bin/admin 3 paxos 1 rsh paxos07 ~/PAXOS/Paxos/paxos/bin/admin 4 paxos 1
36	admin-kill. sh	<u>起動マシン上の起動中のプロセス paxos/bin/adminを調べてKILL</u> (rm-admin-control. sh, rm-admin-paxos. sh からリモート起動される) USAGE: admin-kill. sh

PFSの試用運用のためのTOOL概要

TOOLコマンド	概 要
37 rm-probe-accept. sh	<p><u>各サーバのACCEPT状態の出力</u> USAGE: rm-probe-accept. sh ...全サーバ対象 rm-probe-accept. sh server-ID ...指定サーバ対象 rsh paxos03 ~/PAXOS/Paxos/PFS/bin/PFSProbe 0 accept rsh paxos04 ~/PAXOS/Paxos/PFS/bin/PFSProbe 1 accept rsh paxos05 ~/PAXOS/Paxos/PFS/bin/PFSProbe 2 accept rsh paxos06 ~/PAXOS/Paxos/PFS/bin/PFSProbe 3 accept rsh paxos07 ~/PAXOS/Paxos/PFS/bin/PFSProbe 4 accept</p>
38 rm-probe-lock. sh	<p><u>各サーバのLOCK保持状態の出力</u> USAGE: rm-probe-lock. sh ...全サーバ対象 rm-probe-lock. sh server-ID ...指定サーバ対象 rsh paxos03 ~/PAXOS/Paxos/PFS/bin/PFSProbe 0 lock rsh paxos04 ~/PAXOS/Paxos/PFS/bin/PFSProbe 1 lock rsh paxos05 ~/PAXOS/Paxos/PFS/bin/PFSProbe 2 lock rsh paxos06 ~/PAXOS/Paxos/PFS/bin/PFSProbe 3 lock rsh paxos07 ~/PAXOS/Paxos/PFS/bin/PFSProbe 4 lock</p>
39 rm-server-exitcheck. sh	<p><u>各サーバ(PFSServer) のexit情報履歴出力</u> USAGE: rm-server-exit-status. sh rsh paxos03 cat /tmp/PFSServer-0*. log rsh paxos04 cat /tmp/PFSServer-1*. log rsh paxos05 cat /tmp/PFSServer-2*. log rsh paxos06 cat /tmp/PFSServer-3*. log rsh paxos07 cat /tmp/PFSServer-4*. log</p>
40 rm-server-status. sh	<p><u>各サーバのクライアントの状態情報出力</u> USAGE: rm-server-exit-status. sh ・サーバ起動状況 ・管理ファイル情報 (ls -l PFS/bin/PFS-*) ・サーバEXIT情報</p>
41 rm-print_log_server. sh	<p><u>各サーバのPFSServer のログ情報を出力 (5サーバ対応)</u> USAGE: rm_print_log_server. sh [SID [noflash]] rm_print_log_server. sh ...全サーバ対象、noflushモード 各サーバprint_log_server. shをリモート起動する</p>
42 rm8-print_log_server. sh	<p><u>各サーバのPFSServer のログ情報を出力 (8サーバ対応)</u> USAGE: rm_print_log_server. sh [SID [noflash]] rm_print_log_server. sh ...全サーバ対象、noflushモード 各サーバprint_log_server. shをリモート起動する ※提供ソースでは、COREサーバ5台、EXTENTIONサーバ最大3台構成です SID=0, 1, 2, 3, 4 : CORE =5, 6, 7 : EXTENTION</p>
43 print_log_server. sh	<p><u>各サーバのPFSServer のログ情報を出力</u> USAGE: print_log_server. sh id [noflash]</p>
44 print_log_client. sh	<p><u>PFSCClient のログ情報を出力</u> USAGE: print_log_client. sh logfile_name logfile_name: PFS/bin/CLIENT_LOG/下の xxxx-nnnファイルのxxxxを指定する</p>
45 rcp-source. sh	<p><u>paxos00環境のPFSのソースコードを各サーバ(paxos03/04/05/06)にコピーする。</u> USAGE: rcp-source. sh ※管理構造は、paxos00と全く同じ。bin下のロードモジュール、SHELLスクリプトは対象外。</p>
46 f-diff. sh	<p><u>sample.cと PFS/bin/Rfile-nnn, Tfile-nnn, Ufile-nnnの diffをとる</u> USAGE: f-diff. sh</p>

PFSServer、admin、PFSSProbeのコマンド仕様

PFSServer : PFSServerのコマンド仕様		概 要
1	PFSServer sid directory [autonomic [target-sid]] sid: サーバーID (=0,1,2,3,4,5,...) directory: サーバーデータ管理ディレクトリTOPパス サーバ起動ディレクトリからの相対パス名 autonomic: 自動立上げモード、他の稼働中のサーバーの Snapshotデータを自動的に調べて最善の Snapshotデータを自分のところへコピーして 立上げ処理をする。 target-sid: autonomicと一緒に指定される 立上がり時に使用するSnapshotデータを持つ サーバーID ※PFSServerは、~/PAXOS/Paxos/PFS/binにある	PAXOSファイルシステム(PFS)のサーバプログラム 実行例 1. PFSServer 0 PFS-0 初期化モードで起動 サーバーID =0、 ディレクトリ PFS-0以下でデータ管理 2. PFSServer 0 PFS-0 autonomic 自動立上げモードで起動 サーバーID =0、 ディレクトリ PFS-0以下でデータ管理 3. PFSServer 0 PFS-0 autonomic 1 サーバーID =1のSnapshotデータによる起動 サーバーID =0、 ディレクトリ PFS-0以下でデータ管理
adminコマンド仕様		概 要
1	※adminは、~/PAXOS/Paxos/paxos/binにある admin sid control	指定サーバー(sid)の保持しているサーバーの管理情報を得る
2	admin sid paxos [page]	指定サーバー(sid)の保持しているSnapshot、ページ管理情報などを得る page: Dumpする場合のページ数
3	admin sid {stop restart} tosid	指定サーバー(sid)を停止状態(stop)に、または通常状態に再開(restart)する tosid: メッセージ通知するサーバーID
4	admin sid down	指定サーバー(sid)を終了する
5	admin sid command string no_k	指定サーバー(sid)にコマンドを送信する
6	admin sid recvstop msgid {0 1}	指定サーバー(sid)がコマンド(msgid)を受信するのを停止(1)する、または受信再開(0)する
7	admin sid birth	指定サーバー(sid)以外の最長老サーバをマスターサーバーにするようサーバー(sid)に指示する
8	admin sid suppress DT (DT: sec)	・サンプルクライアントプログラム (PFSSClientMain.c)で 提供されているコマンド
9	admin sid catchupstop {0 1}	指定サーバー(sid)のCatchup処理を停止(0)、再開(1)する
10	admin sid newround	
11	admin sid begin no_k	
PFSSProbeコマンド仕様		概 要
1	※PFSSProbeは、~/PAXOS/Paxos/PFS/binにある PFSSProbe sid accept	指定サーバー(sid)の保持している受信したコマンド情報を得る
2	PFSSProbe sid lock	指定サーバー(sid)の保持しているロック管理情報を得る

※adminの詳細は、~/PAXOS/Paxos/paxos/test/admin.c のプログラムを参照。

※PFSSProbeの詳細は、~/PAXOS/Paxos/PFS/test/PFSSProbe.c のプログラムを参照。

PFSCClientのコマンド仕様

	PFSCClientコマンド仕様	概要
1	PFSCClient session	セッションを開始して何もせず終了する
2	PFSCClient lockW lockname PFSCClient lockW2 lockname sleeptime	Write ロックの取得し、sleep 50秒後にロック解放する lockW2: sleep timeを指定
3	PFSCClient lockR lockname	Read ロックの取得し、sleep 50秒後にロック解放する
4	PFSCClient writeinbound fromFile [toFile] PFSCClient writeinbound2 fromFile [toFile]	ファイルのサイズに関係なくInbound処理でWriteし、 sleep 50秒後に終了する writeinbound2: sleep 50秒待ちなし
5	PFSCClient writeoutbound fromFile [toFile] PFSCClient writeoutbound2 fromFile [toFile]	ファイルのサイズに関係なく Outbound処理でWriteし、 sleep 50秒後に終了する writeoutbound2: sleep 50秒 待ちなし
6	PFSCClient write fromFile [toFile] PFSCClient write2 fromFile [toFile]	ファイルのサイズにより自動的にInbound/Outbound の振り分け処理をしてWriteし、sleep 50秒後に終了する writeoutbound2: sleep 50秒 待ちなし
7	PFSCClient lockwrite fromFile [toFile]	Write ロックを取得してファイルのWrite処理をする lockW + write filename + lockWのUnlock
8	PFSCClient read fromFile [toFile]	ファイルのRead処理
9	PFSCClient lockread fromFile [toFile]	Read ロックを取得してファイルのRead処理をする lockR + read filename + lockRのUnlock
10	PFSCClient delete filename	・サンプルクライアントプログラム (PFSCClientMain.c) で提供
11	PFSCClient ephemeral fromFile [toDirectory] PFSCClient ephemeral2 fromFile [toDirectory] PFSCClient ephemeral3 fromFile [toDirectory]	ephemeral fileの作成し、sleep 50秒後に終了する (Clientプロセス終了でファイル自動削除) ephemeral2: sleep 1秒 ephemeral3: sleep待ちなし
12	PFSCClient mkdir dirname	Directoryの作成
13	PFSCClient rmdir dirname	Directoryの削除
14	PFSCClient readdir dirname	指定Directory下の情報取得
15	PFSCClient eventdir dirname version count PFSCClient eventdir2 dirname version count	指定Directoryに関するevent取得、Directory情報取得 eventdir2: Directory情報の出力なし
16	PFSCClient eventlockW lockname count	指定したWrite lockのlocknameに関するevent取得
17	PFSCClient eventlockR lockname count	指定したRead lockのlocknameに関するevent取得
18	PFSCClient snapshot	snapshotを取得する
19	PFSCClient lockwritelooop fromFile toFileTop nloop (0<nloop<9999)	下記処理を指定回数(nloop)繰り返す (nloop) x (SessionOpen+lockW+write+Unlock+SessionClose)
20	PFSCClient lockwritelooop2 fromFile toFileTop nloop (0<nloop<9999)	下記処理を指定回数(nloop)繰り返す セッションは開始した状態で繰り返し処理をする SessionOpen + (nloop) x (lockW+write+Unlock) + SessionClose
21	PFSCClient lockreadloop fromFile toFileTop nloop (0<nloop<9999)	下記処理を指定回数(nloop)繰り返す (nloop) x (SessionOpen+lockR+read+Unlock+SessionClose)

PFSCClientのコマンド仕様

PFSCClientコマンド仕様		概要
22	PFSCClient lockreadloop2 fromFile toFileTop nloop (0<nloop<9999)	下記処理を指定回数(nloop)繰り返す セッションは開始した状態で繰り返し処理をする SessionOpen + (nloop)x(lockR+read+Unlock) + SessionClose
23	PFSCClient post queueName data	指定したqueueName にデータをポストする (waitで指定したqueueName であること)
24	PFSCClient wait queueName	指定したqueueName の先頭からポストされている データを取得する 取得できるまで永遠に待つ (postで指定したqueueName であること)
25	PFSCClient eventQueue queueName [count]	指定したqueueName に関するevent取得 countが指定されていればcount数のeventを取得

※PFSCClientの詳細は、~/PAX00S/Paxos/PFS/test/PFSCClientMain.c のサンプルクライアントプログラムを参照。