

CLUSTERPRO

MC ProcessSaver 1.0 for Linux

FAQ 集

© 2013(Apr) NEC Corporation

- ☐ 導入に関する質問
- ☐ 運用に関する質問
- ☐ 動作環境に関する質問

はしがき

本書は、CLUSTERPRO MC ProcessSaver 1.0 for Linux（以後 ProcessSaver と記載します）の FAQ について記載したものです。

（1）本書は IA32 搭載マシンの以下のオペレーティングシステムに対応します。

- Red Hat Enterprise Linux AS 3.0
- Red Hat Enterprise Linux AS 4.0
- Red Hat Enterprise Linux ES 3.0
- Red Hat Enterprise Linux ES 4.0
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Oracle Enterprise Linux 5
- Oracle Enterprise Linux 6
- SUSE Linux Enterprise 9
- SUSE Linux Enterprise 10
- SUSE Linux Enterprise 11
- MIRACLE LINUX V3.0
- Turbolinux 10 Server
- Asianux 2.0
- CentOS 4

また、以下の OS は EM64T 搭載マシンにも対応しています。

- Red Hat Enterprise Linux AS 3.0
- Red Hat Enterprise Linux AS 4.0
- Red Hat Enterprise Linux ES 3.0
- Red Hat Enterprise Linux ES 4.0
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Oracle Enterprise Linux 5
- Oracle Enterprise Linux 6

(2) 本書の文中で、特に指定がなければ「ProcessSaver」は「CLUSTERPRO MC ProcessSaver 1.0 for Linux 以降のバージョン」を示すものとします。

(3) 商標および登録商標

- ✓ Red Hat は、米国およびその他の国における Red Hat,Inc.の登録商標または商標です。
- ✓ SUSE は、米国およびその他の国における Novell,Inc. の商標 です。
- ✓ MIRACLE LINUX は、ミラクル・リナックス株式会社の登録商標です。
- ✓ Asianux は、ミラクル・リナックス株式会社の日本における登録商標です。
- ✓ Turbolinux は、ターボリナックス株式会社の商標または登録商標です。
- ✓ Linux は、Linus Torvalds 氏の米国およびその他の国における、登録商標または商標です。
- ✓ その他、本書に登場する会社名および商品名は各社の商標または登録商標です。
- ✓ なお、本書では®、TM マークを明記しておりません。

目次

1	導入に関する質問	1
1.1	設定に関する質問	1
1.2	ライセンスに関する質問	8
2	運用に関する質問	9
2.1	起動、終了に関する質問	9
2.2	監視に関する質問	18
2.3	再起動スクリプトに関する質問	26
3	動作環境に関する質問	29

1 導入に関する質問

1.1 設定に関する質問

Q1.	複数の同一名プロセスを個数によって監視したいのですが、どのような設定をすればよいでしょうか？
A1.	1. pfile のオプション部に min_proc_count を指定することで、監視対象プロセスを含めた同一名プロセス数の監視を行うことができます。 下記に 同名のプロセスが 4 個未満になるとプロセスの消滅を検出して、再起動スクリプトを実行する pfile の記述例を示します。 ○ pfile の例 <pre>##### PENT ##### [プロセス名]:[再起動スクリプト]:86400:3:continue:min_proc_count=4</pre> 2. ただし、1. のように min_proc_count でプロセスの個数監視をしても、pcheck は必ず同名プロセス群から代表して一つのプロセスを監視対象とします。 そのため、この代表として選択された監視対象プロセスが異常終了してしまった場合は、たとえ同名のプロセスが min_proc_count で指定した数以上起動していても、プロセスの異常終了を検出します。 この制限に対応したい場合には、プロセスの個数をカウントするシェルスクリプトを作成し、作成したスクリプトを pcheck で監視する必要があります。 下記に nfsd プロセスの起動済みのプロセス数をカウントし、規定のプロセス数(ここでは 4)より小さくなると異常終了するスクリプトの例と、その際の再起動スクリプトの記述例を示します。 ○ プロセス数をカウントするシェルスクリプトの記述例 <pre>#!/bin/sh # restart shell for Multi proc sample # process name DAEMON_NAME="nfsd" # process alive count</pre>

```

DAEMON_COUNT=4
# process check interval sec
SLEEP_TIME=60

STATUS=0

while [ ${STATUS} -eq 0 ]
do
    daemonfname=`/bin/basename ${DAEMON_NAME}`
    num_daemon=`ps -e | /usr/bin/awk -v fname=${daemonfname} '$4 ==
fname {print $1}' | /usr/bin/wc -l`
    if [ ${num_daemon:=0} -lt ${DAEMON_COUNT} ]
    then
        STATUS=1
    else
        /bin/sleep ${SLEEP_TIME}
    fi
done

return 0

```

○再起動スクリプトの記述例

```

#!/bin/sh
# restart shell for Multi proc sample

[監視対象プロセスの再起動処理]

# waiting for proc wakeup
/bin/sleep 30

# multi process checker start
/var/opt/HA/PS/conf/bin/MULTI_PROC/multi_check.sh &

```

詳細は、「ユーザーズガイド」の「同一名プロセス監視の導入手順」の章を参照してください。

Q2.	起動ごとに引数が変わってしまうプロセスを監視したいのですが、どのような設定をすればよいでしょうか？
A2.	起動ごとに引数が変わらない固定部分を、プロセス名として指定することで監視が可能です。固定部分が同じプロセスが複数ある場合には、同一名プロセス監視を使用することで、監視が可能です。 詳細は、「ユーザーズガイド」の「同一名プロセス監視の導入手順」の章を参照してください。

Q3.	ProcessSaver が出力する syslog のメッセージを停止させたり、ファシリティ・レベルを変更したいのですが、どのような設定をすればよいでしょうか？
A3.	環境変数 HAPS_SILENT_MODE に 1 を設定して ProcessSaver を起動することで、ProcessSaver が出力するすべての syslog を抑制することができます。 ファシリティ・レベルを変更する機能については現在はサポートしておりません。 なお、ProcessSaver が出力する syslog のファシリティとレベルは以下となります。 • ファシリティ: LOG_USER • レベル : LOG_ERR または LOG_WARNING 詳細は、「ユーザーズガイド」の「サイレントモードでの運用手順」の章を参照してください。

Q4.	oracle の監視を行う場合の一般的な設定について教えてください。
A4.	ProcessSaver による oracle 監視は、ApplicationMonitor for Oracle と比べてストール監視等が行えないため推奨しておりません。 ApplicationMonitor for Oracle を導入し、ストール監視・クラスタ連携といった oracle の特性を考慮した監視を導入するのが一般的です。

Q5.	ProcessSaver をインストール後、IP アドレス、ホスト名が変更となった場合、何か処置が必要でしょうか？
A5.	ProcessSaver は IP アドレス・ホスト名に依存していませんので、これらの変更があっても影響はありません。 再起動、再インストール等の処置は不要です。

Q6.	ProcessSaver で用意されている設定ファイル、再起動スクリプトのテンプレートなどがありますか？																			
A6.	ProcessSaver は監視方法を定義する pfile と、監視対象プロセスを自動再開するための手順を記載した再起動スクリプトを用意する必要があります。 pfile・再起動スクリプトは監視対象の各製品プロセスに応じた起動手順や必要な後処理などをお客様で作成し、設定していただく必要があります。 ただし、代表的な製品や OS 標準の重要プロセスなどについてはサンプルやテンプレート集などを用意していますので、それをお使いいただくことで比較的容易に導入していただくことが可能です。 ■ ProcessSaver HP-UX 版 監視テンプレート集 Linux 版 監視テンプレート集 https://www.intra.nec.co.jp/clusterpro/ha/ha_ps/download.html また、特に代表的な Web サーバ製品やアプリケーションサーバ製品の監視については消滅監視とあわせストール監視機能や監視定義の自動生成機能などを用意した専用監視オプション製品も用意しておりますのでご検討ください。 <table border="1"> <thead> <tr> <th>製品名</th><th colspan="2">各種テンプレート</th></tr> </thead> <tbody> <tr> <td rowspan="3">Web サーバ監視オプション ProcessSaver WebEdition</td><td>HP-UX</td><td> Apatch Web Server 監視 Zeus WebServer 監視 Tomcat 監視 </td></tr> <tr> <td>Linux</td><td> Apatch Web Server 監視 Sun One WebServer 監視 Tomcat 監視 </td></tr> <tr> <td>Solaris</td><td> Apatch Web Server 監視 Zeus WebServer 監視 Sun One WebServer 監視 Tomcat 監視 </td></tr> <tr> <td rowspan="2">AP サーバ監視オプション ProcessSaver ApsEdition</td><td>HP-UX</td><td> WebOTX 監視 WebLogic 監視 JBoss 監視 </td></tr> <tr> <td>Linux</td><td> WebOTX 監視 WebLogic 監視 JBoss 監視 </td></tr> <tr> <td>NFS 監視オプション ProcessSaver nfsEdition</td><td>HP-UX</td><td>NFS サーバ&クライアント監視</td></tr> </tbody> </table>		製品名	各種テンプレート		Web サーバ監視オプション ProcessSaver WebEdition	HP-UX	Apatch Web Server 監視 Zeus WebServer 監視 Tomcat 監視	Linux	Apatch Web Server 監視 Sun One WebServer 監視 Tomcat 監視	Solaris	Apatch Web Server 監視 Zeus WebServer 監視 Sun One WebServer 監視 Tomcat 監視	AP サーバ監視オプション ProcessSaver ApsEdition	HP-UX	WebOTX 監視 WebLogic 監視 JBoss 監視	Linux	WebOTX 監視 WebLogic 監視 JBoss 監視	NFS 監視オプション ProcessSaver nfsEdition	HP-UX	NFS サーバ&クライアント監視
製品名	各種テンプレート																			
Web サーバ監視オプション ProcessSaver WebEdition	HP-UX	Apatch Web Server 監視 Zeus WebServer 監視 Tomcat 監視																		
	Linux	Apatch Web Server 監視 Sun One WebServer 監視 Tomcat 監視																		
	Solaris	Apatch Web Server 監視 Zeus WebServer 監視 Sun One WebServer 監視 Tomcat 監視																		
AP サーバ監視オプション ProcessSaver ApsEdition	HP-UX	WebOTX 監視 WebLogic 監視 JBoss 監視																		
	Linux	WebOTX 監視 WebLogic 監視 JBoss 監視																		
NFS 監視オプション ProcessSaver nfsEdition	HP-UX	NFS サーバ&クライアント監視																		

Q7.	複数指定したプロセスのうちどれか一つでも消滅した場合に、全てのプロセスを再起動してプロセス監視を再開させたいのですが、どのような設定をすればよいでしょうか？
A7.	pfile のオプション部に grouptag を指定することで、1 つの pcheck でグループ化して監視することが可能です。 grouptag を使用したグループ監視の pfile の一例 <pre>##### PENT ##### [プロセス a]:[再起動スクリプト 1]:300:3:continue:grouptag=groupXXX [プロセス b]:[再起動スクリプト 2]:300:3:continue:grouptag=groupXXX [プロセス c]:[再起動スクリプト 3]:300:3:continue:grouptag=groupXXX</pre> * groupXXX には、任意のグループ名を指定してください。 グループ名、再起動スクリプト名、GRACE 値、再起動回数、リトライオーバーアクションは同じである必要があります。 上記の例の場合、プロセス a, b, c のどのプロセスの消滅を検知した場合でも再起動スクリプトが実行され、グループ単位に再起動を行うことができます。 再起動スクリプトは、プロセス a, b, c すべての停止・再起動後、sleep 等ですべての対象プロセスが起動するまでの時間、待ち合わせを行う必要があります。 詳細は、「ユーザズガイド」の「グループ監視の導入手順」の章を参照してください。

Q8.	サーバへの導入後にディスク構成が変更となった場合、何か処置が必要でしょうか？
A8.	ProcessSaver は、ディスク構成が変更になった場合でも特に影響はありません。 再起動、再インストールは不要です。

Q9.	PATH の指定にワイルドカードを使用できますか？
A9.	pfile の PATH の記述にワイルドカードを使用することはできません。 同様に pcheck(1M)、padmin(1M) コマンドのファイル名指定でワイルドカードを使用することはできません。使用するファイル名は必ず絶対パスで指定してください

Q10.	監視対象のプロセス自身を、再起動スクリプトとして設定して監視できますか？
A10	ProcessSaver の仕様として、再起動スクリプトは必ず終了する必要があるため、以下のように監視対象のプロセス自身を再起動スクリプトとして設定することはできません。 • PENT の例 xxxx.sh:xxxx.sh:68400:3:continue 対象のプロセスを起動して、スクリプト自体は終了するようなスクリプトを作成して、再起動スクリプトに設定する必要があります。

Q11.	プロセス名が同一でかつプロセスの引数が一部同じであるプロセスを区別して監視したいのですが、どのような設定をすればよいでしょうか？
A11	例えば以下のようなプロセス名のプロセスの監視を行いたい場合。 (1) proc aaaa (2) proc aaaa_bbbb pfile のオプション部に include_strings を指定し、PENT 部を以下の順に設定することで区別して監視を行うことができます。 ProcessSaver は、pfile の PENT 部に記載された順番で監視対象を特定し、一度監視対象と特定されたプロセスについては、他の PENT では監視対象とならないため正しく監視することができます。 ○ pfile の例 <pre>proc::-0:0:continue:include_strings=aaaa_bbbb ★1 proc::-0:0:continue:include_strings=aaaa ★2</pre> ★1 と ★2 を逆に記載した場合は、★2 で監視対象プロセスを判定する際に (1) のプロセスとプロセス名および include_strings に指定された文字列が一致するため、監視対象のプロセスと判断してしまう可能性があります。 必ず上記の順に記載してください。 ただし監視を行いたいプロセスが引数まで同名で複数起動する場合は、本設定でも監視を行うことはできませんので、必ず対象のプロセスが 1 つずつしか起動しないことをあらかじめ確認してください。

Q12.	運用管理製品 (WebSAM SystemManager 等) でログ監視を行う場合に監視対象とすべきメッセージを教えてください。
A12	pcheck が出力するプロセス監視で復旧できないエラーを報告する以下の syslog メッセージを監視してください。 これ以外のメッセージの監視については、特に必須ではありません。 - 消滅監視によるリトライオーバーのメッセージ - RETRY_OVER:CONTINUE - RETRY_OVER:SHUTDOWN - RETRY_OVER:EXIT - 意味と対処 対象プロセスの消滅を検出し、一定回数リトライしても再起動できない状態です。対象プロセスが起動できなかった原因を調査してください。

1.2 ライセンスに関する質問

Q1.	評価用のコードワードから正式コードワードに変更する場合、ProcessSaver で 何か処置が必要ですか？
A1.	ProcessSaver およびマシンの再起動等は特に必要ありません。

Q2.	ProcessSaver でクラスタ構成の場合、ライセンス数は現用系のみでよいですか？それとも現用系／待機系の両方の数が必要ですか？
A2.	現用系／待機系の CPU 数分、必要になります。 また、ライセンス型番にはメディアは含まれませんので、メディアをお持ちで無い場合は、メディア型番1セットも購入が必要となります。

2 運用に関する質問

2.1 起動、終了に関する質問

Q1.	remsh コマンドを使用し、リモートで pcheck を起動しようとしたのですが、コマンドに応答がありません。 何が原因でしょうか？
A1.	remsh コマンドを使用して実行するコマンドをバックグラウンド起動する場合には、リモート側の標準入力／標準出力／標準エラー出力を明示的にクローズする処理を行う必要があります。 クローズしていない場合、リモートで実行するコマンドが終了するまで、remsh は終了しません。そのため、remsh(1) のコマンドラインは下記のようにして実行する必要があります。 # remsh hostname -n "command 1>&- 2>&- &"

Q2.	監視対象のプロセスを起動した後、pcheck を起動しましたが syslog に以下のようなメッセージが出力されて監視が正常に行えません。 何が原因でしょうか？ pcheck[XXX]: Set pid fail (xxx)
A2.	Oracle や Java アプリケーションなどの起動に時間がかかるプロセスを監視する場合、監視対象のプロセスが完全に起動する前に pcheck の監視が開始されてプロセス消滅を検出してしまうことがあります。 監視対象プロセスの起動が確実に終了したことを確認してから pcheck による監視を開始するようにしてください。 スクリプト等で監視対象プロセス および pcheck を自動起動している場合には、以下のような方法が有効です。 - 監視対象プロセスの起動と pcheck 起動の間に、/bin/sleep で監視対象プロセスが完全に起動するまでの時間、確実に待ち合わせを行う。 - pcheck に -w オプションを付与して起動し、pcheck 起動後に一定時間待ち合わせてから監視を開始する。 pcheck を起動して 30 秒後に監視を開始する場合の例 # pcheck -f <pfile 名> -w 30

Q3.	監視対象プロセスを起動した後、pcheck を起動しましたが syslog に以下のようなメッセージが出力されて監視が正常に行えません。 何が原因でしょうか？ pcheck[XXX]: pcheck is already execute. または pcheck[XXX]: shmget(2) use other process.
A3.	pfile の IPCKEY で指定した共有メモリの値が他のプロセスですでに使用されている可能性があります。 IPCKEY をシステムで一意となる 0x00000001～0x7fffffff に変更した後、pcheck の再起動を行ってください。 pfile の例 <pre>##### PARAM ##### IPCKEY 0x10000001 ← ★この値を変更</pre> * IPCKEY の値が小さい場合、他のプロセスが使用する共有メモリとバッティングする可能性が高くなります。このことから IPCKEY の値は 0x1f000000 以降の大きな値を指定するよう推奨します。
Q4.	padmin コマンドで pcheck を終了しようとしたのですが、syslog に以下のようなメッセージが出力されて pcheck が終了しません。pcheck を終了する方法を教えてください。 pcheck[XXX]: Can't shutdown [pcheck]
A4.	kill(1) コマンドを使用して、手動で pcheck を終了してください。

Q5.	rc から pcheck を起動したのですが、起動後しばらくすると pcheck が消滅してしまいます。 何が原因でしょうか？
A5.	rc から pcheck を起動する場合、rc 処理の最後に実行される残存プロセスの強制停止処理を受け付けないように trap 処理を追加する必要があります。 pcheck は、起動後に rc からの強制停止を受けないように初期化処理の中で rc からの強制停止をブロックする処理を実行しています。その初期化処理が実行される前に、rc の強制停止処理を受け pcheck が消滅している可能性があります。 なお、rc の強制停止処理は、すべての rc 起動が終わったあとに実行されます。 このため rc3 からの起動時に発生する可能性が高く、rc3 からの起動時には rc ファイルに trap 処理を追加して、強制停止処理を受けないようにする処理を追加することをマニュアル等で明記しております。 ただし、rc3 に登録されているスクリプトが少ない場合や rc 機能の内部制御処理が短い場合には rc2 からの起動時にも発生する可能性がありますので、このような場合には rc2 からの起動の場合にも trap 処理を追加する必要があります。 <pre>#!/bin/sh # trap "" 1 2 3 13 15 ← rc ファイルにこの行を追加</pre> 詳細は、「ユーザズガイド」の「起動、終了ファイルの導入について」の章を参照してください。

Q6.	pcheck 自身の異常終了をログ等で確認することはできますか？
A6.	pcheck 自身が、ファイル I/O の異常やメモリ不足などで終了した場合は、ProcessSaver のダンプファイルに内部ログが出力されます。 これらを確認することで、pcheck の終了を確認することが可能です。

Q7.	ProcessSaver が他のプロセスに対してシグナルを送信することはありますか？
A7.	ProcessSaver は他のプロセスに対してシグナルを送信することはありません。

Q8.	pcheck が起動できない、もしくは起動するがすぐに終了してしまいます。 何が原因でしょうか？
A8.	syslog に pcheck 関連のメッセージが出力されていないか確認してください。 pcheck 関連のメッセージが出力されている場合、メッセージ単位にそれぞれ以下の内容を確認してください。 1. pfile init fail. pfile = 'pfile 名' ○ pfile 中の各設定に指定できない不正な値が設定されていたり、構文が間違っている等、pfile が不正である可能性があります。 pfile 記述が正しいか確認してください。 2. fopen(3) fail. pfile = 'pfile 名' pfile init fail. pfile = 'pfile 名' ○ 指定したパスに pfile が実際に配置されているか、配置されている場合、コマンドラインのパス指定が誤っていないか確認してください。 ○ また、root ユーザー以外で pcheck を実行する場合は、パーミッションや所有権によって、pfile の読み込みに失敗する場合があります。 起動ユーザーの READ 権限、所有権を確認してください。 3. pcheck is already execute. shmget(2) fail. errno = 'エラーNo' ○ 指定された pfile 名で pcheck がすでに起動しているか他の pfile と IPCKEY (共有メモリ) の値が重複している可能性があります。 padmin -l コマンドで、同名の pfile 名で pcheck が起動していないことを確認してください。 4. shmget(2) fail. errno = 'エラーNo' ○ 指定された IPCKEY (共有メモリ) の値が他のプロセスと重複している可能性があります。IPCKEY の値をシステム上で一意の値に変更してください。 現在使用されている共有メモリの一覧は、ipcs(1) コマンドで確認することができます。 5. illegal option. ○ pcheck 起動時に無効なオプションが指定されました。 同時に出力される Usage に従って正しいコマンドを指定してください。

Q9.	pcheck が syslog 等のメッセージ出力なしに突然終了してしまいます。 何が原因でしょうか？
A9.	通常 pcheck の終了時には何らかのメッセージを syslog に出力します。 メッセージの出力もなく、ProcessSaver の共有メモリダンプファイルも出力せずに終了するケースとしては、外部からのシグナル受信による強制停止が考えられます。 オペレータからの kill(1) コマンドによるシグナル送信以外の原因としては、pcheck を起動するためにミドルウェア製品 (JobCenter や NNM) を利用していた場合、pcheck の起動スクリプトの終了時に、起動したプロセスに対してシグナルを送信して残っているプロセスの回収処理を行っている事例がありました。 この場合、以下の回避策で対応してください。 1. pcheck を起動しているミドルウェア製品の設定などを変更し、上記回収処理を行わないようにする。 2. pseparate (ProcessSaver が用意しているプロセスグループ分離コマンド) を使用して pcheck を起動するように変更する。 • 起動スクリプトの pcheck 起動時の記述例 /opt/HA/PS/bin/pseparate /opt/HA/PS/bin/pcheck -f <pfile 名>

Q10.	padmin -l コマンドを実行しても、何も表示されません。 何が原因でしょうか？
A10.	まず、ps(1) コマンドを実行し起動中の pcheck が表示されるか確認してください。 (例) # ps -ef grep pcheck → pcheck プロセスが表示されるか確認 表示されない場合、プロセス監視が起動していないか、 起動後すぐに終了している可能性があります。 原因については、起動、終了に関する質問の Q8 または Q9 を 確認してください。 表示されるにも関わらず padmin -l コマンドで一覧表示されない場合には、 ProcessSaver の管理ファイルが削除されている可能性があります。 /var/tmp 配下の 管理ファイルを削除していないか確認してください。 (例) # ls -la /var/tmp → .pcheck.XXXX ファイルが存在するか確認 管理ファイルが削除されている場合には、padmin -l コマンドでの一覧表示を 行うことができません。ただしプロセス監視自体にはまったく影響はございま せん。 また、この状態は以下のどちらかの手順を実行することで復旧可能です。 - padmin コマンドで reload を実行する。 # padmin -f <pfile 名> -c reload - pcheck を再起動する。

Q11.	pcheck を複数起動しているシステムで、サーバ停止に時間がかかってしまいます。 何が原因でしょうか？
A11.	通常、サーバ停止時には rc ファイル内で、「padmin -f <pfile 名> -c shutdown」を実行して pcheck の停止処理が実行されます。 * padmin コマンドは、pcheck の動作を予約するコマンドで pcheck プロセスとの同期を取る必要があるため、pfile の MONITOR_INTERVAL に設定された秒数、実行を待ち合わせて終了する仕様となっています。 つまり、rc スクリプトを pcheck 単位に複数登録しているシステムの場合、サーバ停止時には pcheck の個数 × MONITOR_INTERVAL(秒) の時間がかかることになります。 これらを平行で実行することで、シャットダウン時の停止時間を短縮することができます。 OS シャットダウン時の pcheck の停止処理を平行に実行するには、現在フォアグラウンドで実行している、padmin の shutdown コマンドを バックグラウンドで実行し、一定時間 pcheck の終了を待ち合わせるように変更します。 ※ 待ち合わせ時間はすべての pcheck が停止するのに十分な時間必要となりますが、目安としては pfile の MSG_CHECK_INTEVAL に 指定してある秒数の 2 倍となります。 デフォルト値は 5 秒となっていますので、以下の例では 10 秒を指定しています。 [変更前] pcheck 起動・停止 rc ファイル中 # /opt/HA/PS/bin/padmin -f <pfile 名> -c shutdown [変更後] # /opt/HA/PS/bin/padmin -f <pfile 名> -c shutdown & # /bin/sleep 10 また、rc 経由で複数の pcheck を起動・終了している場合、各 pcheck の rc ファイルを一つにまとめる必要があります。 具体的な設定変更手順は以下の通りです。 1. rc 経由で起動・停止している pcheck 群の rc ファイルを一つにまとめ、ひとつの rc ファイルからすべての pcheck を起動・停止するように変更する。 2. 停止処理中の padmin -f <pfile 名> -c shutdown コマンドをフォアグラウン

ド起動からバックグラウンド起動に変更する。

3. すべての padmin -f <pfile 名> -c shutdown コマンド投入後 pcheck 終了を待ち合わせるため MSG_CHECK_INTERVAL ×2 の秒数 sleep コマンドで待ち合わせる。

----- 複数の rc スクリプトをひとつにまとめた場合の例 -----

<省略>

```
PCHECK=/opt/HA/PS/bin/pcheck
PADMIN=/opt/HA/PS/bin/padmin
PFILE1=/var/opt/HA/PS/conf/bin/pfileA
PFILE2=/var/opt/HA/PS/conf/bin/pfileB
PFILE3=/var/opt/HA/PS/conf/bin/pfileC
PFILE4=/var/opt/HA/PS/conf/bin/pfileD
```

```
case $1 in
```

```
start)
```

```
    (${PCHECK} -f ${PFILE1} &) >/dev/null 2>&1
```

```
    (${PCHECK} -f ${PFILE2} &) >/dev/null 2>&1 ★ pcheck 起動を複数記載
```

```
    (${PCHECK} -f ${PFILE3} &) >/dev/null 2>&1 ★ "
```

```
    (${PCHECK} -f ${PFILE4} &) >/dev/null 2>&1 ★ "
```

```
    ;;
```

```
stop)
```

```
    ${PADMIN} -f ${PFILE4} -c shutdown & ★ pcheck 停止をバックグラウ  
ンドで複数記載
```

```
    ${PADMIN} -f ${PFILE3} -c shutdown & ★ "
```

```
    ${PADMIN} -f ${PFILE2} -c shutdown & ★ "
```

```
    ${PADMIN} -f ${PFILE1} -c shutdown & ★ "
```

```
    /usr/bin/sleep 10 ★ padmin の終了を sleep で待ち合  
わせ
```

※ sleep に指定する値は MSG_CHECK_INTERVAL の 2 倍を指定

```
    ;;
```

<省略>

----- 複数の rc スクリプトをひとつにまとめた場合の例 -----

Q12.	OS 停止時に ProcessSaver の core ファイルが作成されてしまいます。何が原因でしょうか？
A12.	OS の起動・停止と連携して pcheck を起動・停止する場合、正しく rc ファイルを作成、設定していないと core ファイルが作成される場合があります。 本件は、OS 停止処理でのみ発生する場合がありますが、通常運用中には発生することはありません。 また core ファイルが作成された場合であっても OS 停止の延長で結果的に pcheck は停止されるため運用上問題はありません。 ● 以下の内容について、正しく設定されていることを確認してください。 • rc に pcheck の終了処理を設定していること。 • rc に pcheck の終了処理を設定している場合、pcheck 停止後に監視対象プロセスを停止する設定になっていること。 rc 設定手順の詳細については「ユーザズガイド」の「起動、終了ファイルの導入について」の章を参照してください。

2.2 監視に関する質問

Q1.	プロセス監視を <code>padmin -c stop</code> コマンドで停止し、監視対象プロセスを再起動した後、 <code>padmin -c start</code> コマンドでプロセス監視を再開したらプロセスの消滅を検出しました。何が原因でしょうか？
A1.	pcheck はプロセス ID(PID) をキーにプロセスの消滅監視を行いますので、監視対象プロセスを再起動した場合など、PID が変更された場合は、プロセス監視の再開 (start) 前に再読み込み (reload) が必要です。 また、padmin コマンドの処理要求が実行されるまでには最大で MSG_CHECK_INTERVAL に指定した秒数必要となりますので、reload を実行してから監視を再開 (start) するまでには、念のため MSG_CHECK_INTERVAL の二倍の秒数待ち合わせてください。 <pre># padmin -f <pfile 名> -c reload</pre> ↓ 最低 MSG_CHECK_INTERVAL の2倍の秒数待ち合わせ <pre># padmin -f <pfile 名> -c start</pre>

Q2.	プロセス監視を <code>padmin -c stop</code> コマンドで停止し、監視対象プロセスを再起動した後、 <code>padmin -c reload</code> コマンドで再読み込み後すぐに <code>padmin -c start</code> コマンドでプロセス監視を再開したらプロセス消滅を検出しました。何が原因でしょうか？
A2.	padmin コマンドは処理を受け付けるだけで、実際に処理 (reload) が実行されるまでには、最大で MSG_CHECK_INTERVAL に指定した秒数かかります。 したがって、安全のため、reload を実行してから監視を再開 (start) するまでに、MSG_CHECK_INTERVAL の設定値の二倍程度の時間待ち合わせてください。 <pre># padmin -f <pfile 名> -c reload</pre> ↓ 最低 MSG_CHECK_INTERVAL の2倍の秒数待ち合わせ <pre># padmin -f <pfile 名> -c start</pre>

Q3.	pcheck を階層的に使用したグループ監視で、親の pcheck の再起動を行うことは可能ですか？
A3.	ProcessSaver では、グループ監視で親の pcheck が終了した場合に、再起動を行う機能はありません。 ただし、親の pcheck の監視・再起動は、/etc/inittab に登録することで可能です。詳細は、inittab(5) のオンラインマニュアルを参照してください。 /etc/inittab に pcheck を登録する際の注意事項は以下となります。 • クラスタウェアで親の pcheck を登録する場合は、pcheck は クラスタウェア から起動されますので、クラスタウェア 以外から再起動は行わないようにしてください。 • inittab に登録された親の pcheck は終了しません。 このため、サーバ停止時に子の pcheck の消滅を検出してしまう場合がありますので子プロセス終了時に、親プロセスによる監視を一時停止する必要があります。 <pre>##### 子プロセスの rc ファイル停止処理の指定例 ##### : 'stop') /opt/HA/PS/bin/padmin -f <親の pfile 名> -c stop ★親プロセスの監視を一時停止 /bin/sleep 5 ★ 念のため sleep にて待ち合わせ /opt/HA/PS/bin/padmin -f <子の pfile 名> -c shutdown ★ 子プロセスの終了 ;; ##### 子プロセスの rc ファイル停止処理の指定例 #####</pre>

Q4.	ProcessSaver のプロセス自身 (pcheck) を監視することは可能ですか？
A4.	グループ監視のように、pcheck から pcheck を階層的に監視することができます。 ただし、階層的に pcheck の監視を行った場合も、最上位の pcheck については監視することはできません。

Q5.	ProcessSaver では、個々のプロセスの生死状態をどのように監視していますか？
A5.	/proc ファイルシステムを使用して、プロセスの死活監視を行なっています。 そのため ps コマンドのように、OS 負荷等による影響を受けることはありません。

Q6.	pcheck 起動時、監視はいつから開始されるのですか？
A6.	pcheck 起動後、MONITOR_INTERVAL で設定した時間(秒)が経過してから、監視が開始されます。 pcheck 起動時および reload 実行時に監視対象プロセスが起動していない場合は、監視対象プロセスが起動するまで一定時間自動的に待ち合わせを行います。 この待ち合わせ時間はデフォルトでは 60 秒となっておりますが、環境変数 (HAPS_PENDING_TIME) を設定することで、待ち合わせ時間を変更することが可能となります。 詳細は、「ユーザズガイド」の「pcheck 起動時の自動待ち合わせ時間を変更する手順」の章を参照してください。

Q7.	JobCenter の常駐プロセス (cjcpw) を監視したいのですが、起動後すぐに消滅してしまいます。 どのような設定をすればよいでしょうか？
A7.	cjcpw コマンドは、-c オプションを付けた場合には、JobCenter のプロセスを起動後すぐに消滅します。 -c オプションを指定せずに cjcpw コマンドを実行することで、監視が可能となります。 また JobCenter を監視する手順として、以下の 2 種類があります。 • JobCenter の監視事例 1. cjcpw で起動 ○ cjcpw プロセスのみを ProcessSaver で監視します。 - cjcpw プロセスが常駐し、自身の起動した JobCenter の各プロセスを監視し、異常を検出すれば cjcpw が異常終了します。 2. cjcpw -c で起動 ○ cjcpw プロセスは上記 JobCenter の各プロセスを起動した後すぐに終了します。従って JobCenter プロセスの各々を ProcessSaver で監視する必要があります。 • * バックグラウンドで実行させる様にしないと、シェルスクリプトに制御が戻りません。

Q8.	pfile の include_strings オプションに、文字列 "=" (イコール) を指定することはできますか？
A8.	pfile のオプション部に "=" を指定することはできません。 ただし "&" 区切りで複数列記載することでアンド条件として指定することは可能です。

Q9.	pcheck コマンドを -w オプションを指定して起動した場合、pcheck プロセス自身はすぐに起動しますか？
A9.	例えば "pcheck -f <pfile 名> -w 30" を実行した場合、pcheck プロセスはすぐに起動し監視の開始のみ 30 秒の待ち合わせを行います。 pcheck 起動後、30 秒待ってから監視を開始する場合 <pre># /opt/HA/PS/bin/pcheck -f <pfile 名> -w 30</pre>

Q10.	監視対象プロセスが存在しない場合や消滅を検出した場合に、syslog へ出力されるメッセージについて教えてください。
A10.	対象プロセスが存在しない場合に出力されるメッセージは下記の 2 種類となります。 - Process [xxx, pid=yyy] Down ProcessSaver が監視対象プロセスの起動を確認 (Up を検出) 後に該当のプロセスが消滅した場合に出力されます。 - Set pid fail (xxx) ProcessSaver が監視を開始した時に、プロセスが存在しない場合に出力されます。

Q11.	リトライ回数を、指定されたプロセス名ごとにリセットすることはできますか？
A11.	指定されたプロセス名ごとにリトライ回数をリセットする方法はありません。 padmin -c reload コマンドを実行することで pfile を再読み込みして、定義されたすべてのプロセスのリトライ回数をクリアすることは可能です。 例 # padmin -f <pfile 名> -c reload

Q12.	GRACE 値 (grace_time) は、いつからの秒数ですか？
A12.	GRACE 値は、1回目のプロセスダウンからの秒数となります。 GRACE 時間経過後にプロセスの消滅を検知した場合には、その時刻が GRACE 値の開始タイミングとなり、リトライ回数は 1 に戻ります。 - ドキュメントより抜粋 - GRACE 値とは GRACE 値とはリトライ回数に達するまでの監視時間です。 指定値は 0 秒 ~ 60*60*24*360 秒(約 1 年)の範囲です。 デフォルト値は、10*60 秒です。 GRACE 値で設定した時間内に、リトライ回数に達した場合、リトライオーバーアクションの設定にしたがって処理をおこないます。 GRACE 値で設定した時間内にリトライ回数に達しなかった場合、GRACE 値に到達後の最初のプロセス再開を 1 回目のリトライとして処理をおこないます。 なお、再起動が不要な場合は、0 (ゼロ) を指定してください。 (例) 以下のように pfile を設定した場合 [プロセス名]:[再起動スクリプト名]:600:5:continue 600 秒以内に 5 回再起動されたときに、リトライオーバーアクションを実行します。 5 回再起動を行う間に 1 回目のプロセスダウンから 600 秒を越える場合には、アクションは実行されません。

Q13.	padmin -l コマンドで、pcheck の一覧表示を行ったが、起動中のはずの pcheck が表示されません。 何が原因でしょうか？
A13.	/var/tmp 配下の一時ファイルが削除されている可能性があります。 pcheck は、起動時に /var/tmp 配下に管理ファイルを作成して、pcheck の管理を行っています。 フォーマットは /var/tmp/.pcheck.<管理番号>です。 一度作成されると明示的に消去しない限り、削除されることはありません。 padmin -l コマンドで表示される pcheck 一覧は、この管理ファイルをキーに共有メモリを取得して表示しますが、このファイルが pcheck 起動中に消去されると、padmin -l コマンドで一覧には表示されないが、pcheck が稼働している事象が発生します。 pcheck を再起動、または padmin -f <pfile 名> -c reload コマンドを投入すると、管理ファイルを再作成します。 なお、この管理ファイルが消去されても、pcheck のプロセス監視には影響はありません。 /var/tmp のファイル掃除機能等で、この管理ファイルを削除していないかをご確認願います。

Q14.	すべての監視を停止した状態で、pcheck を起動することはできますか？
A14.	以下のような手順で行うことができます。 1. ProcessSaver の起動前に、あらかじめ一時ファイルを作成しておきます。 # touch /opt/HA/PS/bin/pcheck.ignore 2. ProcessSaver を起動すると、syslog に監視停止中のメッセージが表示されます。 Jun 8 16:23:45 node1 syslog: pcheck wait... by /opt/HA/PS/bin/pcheck.ignore 3. 監視を再開する場合は一時ファイルを削除します。 # rm /opt/HA/PS/bin/pcheck.ignore ただしこの場合、ProcessSaver はすべて一時停止状態で起動されます。 複数の pfile で、ProcessSaver を複数起動している場合は、すべての監視が停止状態で起動されますので注意が必要です。

Q15.	監視停止用の一時ファイル (/opt/HA/PS/bin/pcheck.ignore) は pcheck 停止前に作成しても問題ないですか？
A15.	停止前に作成してもかまいません。 また運用中に作成しておけば次回の pcheck 起動時から有効となります。 監視停止用の一時ファイルは一度作成すると、毎回起動時に作成する必要はありませんので rm コマンド等でファイルを削除しない限り、監視を停止し続けることになります。

Q16.	ProcessSaver のストール監視設定時、pcheck の起動後すぐにストールを検出してしまいます。 何が原因でしょうか？
A16.	ProcessSaver のストール監視は、指定されたファイルのタイムスタンプと現在の時刻とを比較してチェックしております。 初期起動時に古いタイムスタンプのストール監視の対象のファイルが残っている場合にストールと判断してしまう場合があります。 これを防止するために、以下のように pfile の INIT 処理で touch コマンドを使って明示的にファイルのタイムスタンプを更新する処理を指定することを推奨しております。 (例) <pre> _bi_stall { PS_INIT AP:/usr/bin/touch <ストール監視対象ファイル名>: DIRECT:-:-:0 ★追加 PS_EXEC DLL:/opt/HA/PS/lib/lib_bi_stall.sl:DIRECT:-:60:-:2:<ストール監視対象ファイル名>:180 PS_ACTION DLL:/opt/HA/PS/lib/lib_bi_stall.sl:DIRECT:-:-:1:0 } </pre> 詳しくは「ユーザーズガイド」の「ストール監視の導入手順」→「INIT エントリの定義」の章を参照してください。

Q17.	padmin コマンド実行時に以下のエラーが syslog に出力されました。 何が原因でしょうか？ padmin: lockfile open(2) fail. lockfile = <ロックファイル名>. または padmin: lockf(F_LOCK) fail. errno = 9.
A17.	ProcessSaver(pcheck) の起動ユーザーと異なるユーザーで padmin コマンドを実行した場合に出力されます。 これは、pcheck の実行ユーザーで作成されたロックファイルに対するアクセス権が padmin コマンド実行時のユーザーにないために、アクセスに失敗したことを示すメッセージとなります。 padmin コマンドは、pcheck 起動時と同じユーザー（一般的にはどちらも root ユーザー）で実行してください。

2.3 再起動スクリプトに関する質問

Q1.	pfile の再起動スクリプトに su(1) コマンドを指定することは可能ですか？
A1.	pfile の再起動スクリプトに、su(1) コマンドを直接指定することは可能です。 ただし、以下の点に注意してください。 - pfile の再起動スクリプトには、基本的にスクリプトを指定する仕様となっておりますので、su(1) 等のコマンドを直接指定する場合には必ず動作検証を行った上で問題ないことをご確認の後設定してください。 * 一般的には su コマンドを実行する場合、su コマンドを実行するスクリプトを作成し、それを再起動スクリプトに指定する方式を使用しますので、ご検討ください。 - su コマンドを使用する場合、必ず -c オプションを使用してください。 /bin/su - <ユーザー名> -c "<再起動スクリプト名>" - su コマンドは絶対パスで指定してください。 /bin/su - su コマンドを使用してスクリプトを実行する場合、実行するスクリプトに引数を指定することはできません。 /bin/su - <ユーザー名> -c "<再起動スクリプト名> <再起動スクリプトの引数>" * このような場合には、su コマンドを使用してスクリプトを実行するスクリプトを作成し、それを再起動スクリプトとして記載する方式をご利用ください。 <pre>##### スクリプトの作成例 ##### </var/opt/HA/PS/conf/bin/su_start.sh> ----- #!/bin/sh /bin/su - <ユーザー名> -c "<再起動スクリプト名> <再起動スクリプトの引数>" exit 0 ----- ##### スクリプトの作成例 ##### ##### pfile の指定例 ##### proc1:/var/opt/HA/PS/conf/bin/su_start.sh: ... ##### pfile の指定例 #####</pre>

Q2.	対象プロセス自身が自動で再起動するプロセスの場合、再起動スクリプトには何を指定すればよいですか？
A2.	何もしないダミーのスクリプトを再起動スクリプトに指定してください。 ただしダミースクリプトでは、監視対象が再起動されるのに十分な時間を待ち合わせる必要があります。 再起動スクリプトの記述例 <pre>#!/bin/sh export LANG=C /bin/sleep 60 ★ 十分な時間待ち合わせ exit 0</pre>

Q3.	プロセスの消滅を検出後、再起動スクリプトが正しく実行できません。 何が原因でしょうか？
A3.	再起動スクリプトが単体で実行できるか確認してください。単体で実行できる場合、以下の点を確認してください。 - 再起動スクリプトに実行権が付与されていること - 再起動スクリプトが正常終了した場合は、ステータスが必ず 0 となるように最後に exit 0 を記述してください。 - 再起動スクリプト中のコマンドへの PATH が通っていること - PATH が通っていない場合 ・ コマンドを絶対パスで記述する ・ 再起動スクリプトの先頭で環境変数 PATH を設定する - 対象プロセスの起動に時間がかかる場合、対象プロセスより先に pcheck が起動して再起動に失敗したと認識してしまう場合があります。 監視対象プロセスの起動に時間がかかる場合、監視対象プロセスの起動完了まで、pcheck が先に監視を再開していないように、対象プロセスの再起動後には十分な時間、sleep で待ち合わせを行ってください。

Q4.	再起動スクリプトの実行に失敗した場合の syslog メッセージについて、意味と対処を教えてください。
A4.	再起動スクリプト失敗時の代表的な syslog メッセージは以下のようなものがあります。 1. 再起動スクリプトが存在しない場合 pcheck[xxx]: -> execvp(2) fail (2) pcheck[xxx]: -> restart shell fail (exit status) (233) 再起動スクリプトに指定したパスにスクリプトが存在するか確認してください。 2. 再起動スクリプトに実行権が無い場合 pcheck[xxx]: -> execvp(2) fail (13) pcheck[xxx2]: -> restart shell fail (exit status) (233) 再起動スクリプトに実行権が付与されているかどうか確認してください。 3. 再起動スクリプトの構文エラーの場合 pcheck[xxx]: -> restart shell fail (exit status) (127) 再起動スクリプトの構文が正しくない可能性があります。 再起動スクリプトが単体で実行できるか確認してください。

3 動作環境に関する質問

Q1.

ProcessSaver を動作させるために必要なディスク使用量はどれくらいですか？

A1.

ProcessSaver は /opt と /var 配下を使用します。使用量は以下の通りです。

マウントポイント	説明	サイズ
/opt 配下	デーモンプロセスやコマンド等のバイナリファイル	約 1MB
/var 配下	デーモンプロセスやコマンド等のログファイル	約 1MB

ログファイルはすべてサイクリックになっていますが、上記は ProcessSaver を一つ起動した場合の使用量です。

pcheck を複数起動する場合、ProcessSaver 一つにつき /var 使用量は 1MB 程度増加します。

Q2.	ProcessSaver が作成する一時ファイルはありますか？
A2.	ProcessSaver は、/var/tmp 配下に以下の一時ファイルを作成します。 /tmp 配下にファイルを作成することはありません。 /var/tmp/.pcheck.(IPCKEY) * 本ファイルは ProcessSaver の管理ファイルとなりますので、削除しないでください。

Q3.	/var 配下の空き容量が無くなった場合、ProcessSaver の動作上なにか問題がありますか？
A3.	空き領域が無い状態では正しく動作しない可能性があるため、動作の保障ができません。 /var 配下の不要ファイルを削除して、空き領域を作成した後 ProcessSaver を再起動してください。 その際、以下のファイルは ProcessSaver の管理ファイルとなりますので削除しないでください。 /var/tmp/.pcheck.(IPCKEY)

Q4.	ProcessSaver のメモリ使用量はどれくらいですか？
A4.	固有メモリ領域は、2M バイト 程度です。 また、共有メモリを 1～4 M バイト使用します。 * 上記は ProcessSaver を一つ起動した場合の使用量です。 pcheck を複数起動する場合は、上記×pcheck の個数となります。

CLUSTERPRO
MC ProcessSaver 1.0 for Linux
FAQ 集

2013 年 4 月 第 2 版

日本電気株式会社
東京都港区芝五丁目7番地1号
TEL (03) 3454-1111 (代表)



© NEC Corporation 2013

日本電気株式会社の許可なく複製、改変などを行うことはできません。
本書の内容に関しては将来予告なしに変更することがあります。

保護用紙