

CLUSTERPRO

MC ProcessSaver 2.2 for Linux

はじめての ProcessSaver

© 2017(Apr) NEC Corporation

- ☐ はじめに
- ☐ プロセス監視の流れ
- ☐ 監視の設定と実践
- ☐ おわりに

改版履歴

版数	改版	内容
1.0	2015.03	新規作成
2.0	2016.03	MC 2.1 に対応
3.0	2017.04	MC 2.2 に対応

はしがき

本書は、CLUSTERPRO MC ProcessSaver 2.2 for Linux (以後 ProcessSaver と記載します)によるプロセス監視の基本的な動作について記載したものです。

(1) 商標および商標登録

- ✓ Linux は、Linus Torvalds 氏の米国およびその他の国における商標または登録商標です。
- ✓ CLUSTERPRO は、日本電気株式会社の登録商標です。
- ✓ ProcessSaver は、日本電気株式会社の登録商標です。
- ✓ 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 マークを明記していません。

目次

1. はじめに	1
2. プロセス監視の流れ	2
3. 監視の設定と実践	3
3.1. 本製品のインストール	3
3.2. 設定ファイルの作成	4
3.3. プロセス監視の開始	9
3.4. 監視状態の確認	10
3.5. プロセスの消滅検出および再起動の確認	12
3.6. リトライオーバーの確認	14
3.7. プロセス監視の復旧	16
3.8. プロセス監視の終了	18
4. おわりに	19

1. はじめに

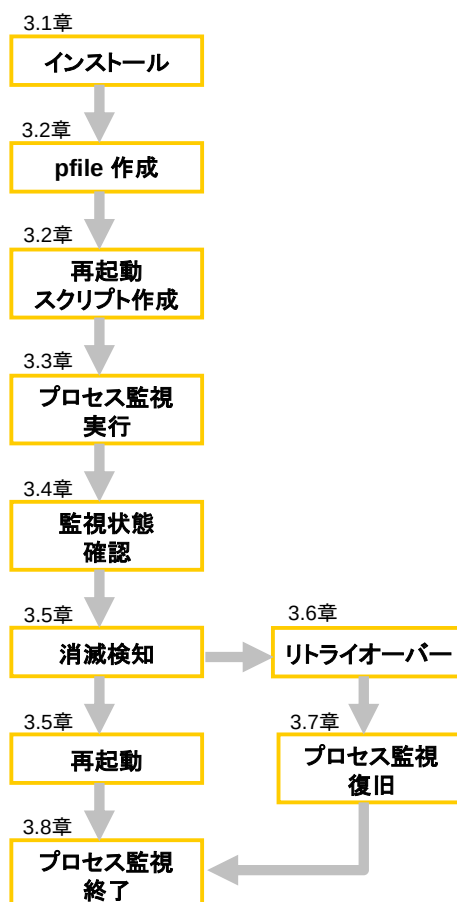
本書では、ProcessSaver によるプロセス監視の基本的な動作についてご説明します。
プロセス監視の設定および監視手順を一連の流れでご確認いただくことで、ProcessSaver の動作を理解することができます。

本書では、以下のような監視例を使用します。

- ・**crond プロセスの消滅を監視**
- ・**crond プロセスが消滅した場合、crond プロセスを自動再開**

なお、各パラメーターの意味や詳細説明につきましては、「CLUSTERPRO MC ProcessSaver クイックリファレンス」をご覧ください。

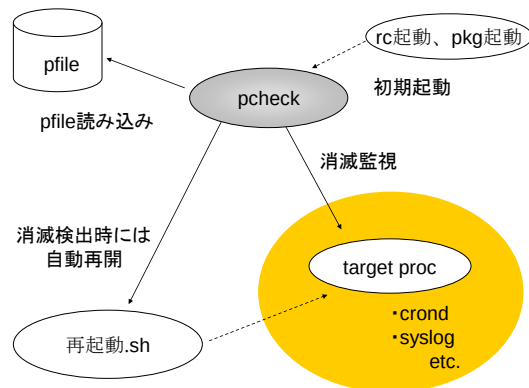
本書では以下のような手順で crond プロセスの監視を行います。



2. プロセス監視の流れ

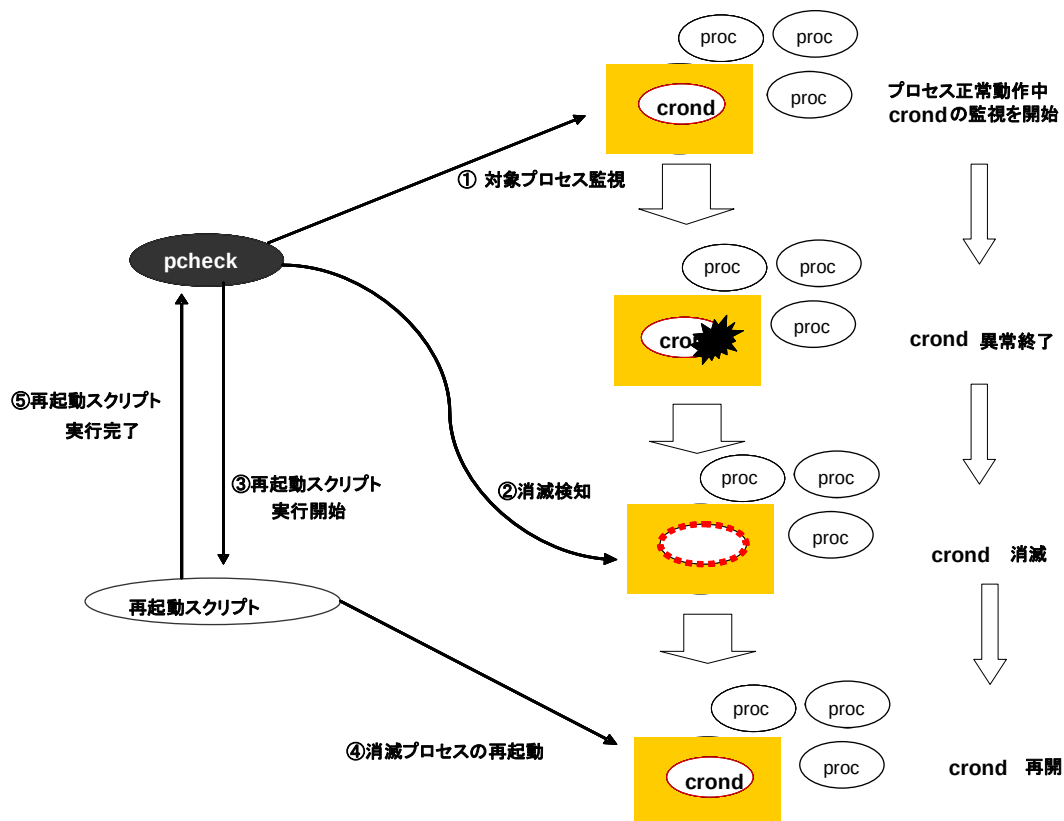
ProcessSaver で提供するプロセス監視の流れをご説明いたします。

ProcessSaver の一般的なプロセス監視の仕組みは以下の図のようになります。



本書でご確認いただく crond プロセスの監視は以下のような流れになります。

- ① プロセス監視を実行する pcheck を起動します。
- ② プロセスの動作状態を一定間隔でチェックし、プロセスの異常終了などを検出します。
- ③ ~ ⑤ 指定された再起動スクリプトにより、対象プロセスを再起動します。



3. 監視の設定と実践

それでは、一緒に ProcessSaver を使用してみましょう。

3.1. 本製品のインストール

まずは ProcessSaver をインストールしましょう。

本製品は以下の手順でインストールします。

コードワードおよびインストールに関する詳細につきましては、「CLUSTERPRO MC ProcessSaver 2.2 for Linux リリースメモ」をご覧ください。

※すでに本製品をインストール済みの場合は、以下の手順を行う必要はありません。
次のステップ(3.2 章)へお進みください。

1. rpm(8)コマンドにより、本製品のパッケージをインストールします。

※以下の手順は、CD-R 媒体を CD-ROM(DVD)装置にマウントした場合の例です。

- 32bit OS(x86 機種)の場合

```
# rpm -ivh /media/Linux/rpm/clusterpro-mc-ps-w.x.y-z.i386.rpm
```

1

- 64bit OS(x86_64 機種)の場合

```
# rpm -ivh /media/Linux/rpm/clusterpro-mc-ps-w.x.y-z.x86_64.rpm
```

1

2. rpm(8)コマンドにより、本製品が正しくインストールされたことを確認します。

```
# rpm -qa | grep clusterpro-mc-ps  
clusterpro-mc-ps-w.x.y-z.*****
```

2

(注) ProcessSaver のバージョンに応じて、バージョン番号の w、x、y、z が更新されます。

***** 部分は OS のバージョンおよびインストールしたパッケージの名称に依存します。

3.2. 設定ファイルの作成

ProcessSaver で使用する設定ファイルを作成しましょう。

ProcessSaver は以下のファイルを使用します。

- ・監視の設定を記述した設定ファイル(以後、pfile と呼びます)
- ・対象プロセスが消滅した場合に再起動を行うスクリプト
(以後、再起動スクリプトと呼びます)

※ProcessSaver は root 権限で実行します。



- pfile を作成しましょう

pfile は pcheck 単位に定義するファイルです。

プロセスの消滅を検出した場合には、pfile に指定された再起動スクリプトによってプロセスを再起動します。再起動スクリプトを指定しない場合は、プロセスが消滅しても再起動は行いませんが、syslog にプロセスが消滅したことを示すメッセージが出力されます。

1. サンプルがあるディレクトリに移動します。

※本製品をインストールすると、自動的にディレクトリが作成され、サンプルは
/var/opt/HA/PS/conf/src 配下に置かれます。

```
# cd /var/opt/HA/PS/conf/src/SG_sample
```

1

2. サンプル pfile を名前を変更してコピーします。

```
# cp pfile /var/opt/HA/PS/conf/bin/pfile_crond
```

2

※使用する pfile はどこのディレクトリ配下に置いても使用できますが、通常は
/var/opt/HA/PS/conf/bin 配下に置いて使用します。

3. コピーした pfile を vi で開き、以下のように編集します。

```
# vi /var/opt/HA/PS/conf/bin/pfile_crond
```

3

```
pfile < /var/opt/HA/PS/conf/bin/pfile_crond >
```

```
##### PARAM #####
IPCKEY                0x1f000501 →①
MSG_CHECK_INTERVAL    5
MONITOR_INTERVAL      10
SHM_DUMP_FILE         /var/opt/HA/PS/log/pcheck_dump_crond →②

##### PENT #####
crond:/var/opt/HA/PS/conf/bin/restart_crond.sh:86400:3:continue
③ ④ ⑤ ⑥ ⑦
```

PARAM 以下は、pcheck 全体の動作を規定する共通情報を指定する部分です。
以下の値を変更してください。

- ① IPCKEY(共有メモリの key)=システム上で一意の値に設定します。
ここでは **0x1f000501** に設定します。
- ② SHM_DUMP_FILE(ダンプファイル名)=システム上で一意の値に設定します。
ここでは **/var/opt/HA/PS/log/pcheck_dump_crond** に設定します。

PENT 以下は、プロセス単位の管理情報を規定する個別情報を指定する部分です。

以下の値を「:(コロン)」で区切って指定します。

- ③ 監視対象のプロセス名 = ここでは `crond` を指定します。
- ④ ③の消滅を検出した場合に実行される再起動スクリプト名
= `/var/opt/HA/PS/conf/bin/restart_crond.sh` に設定します。
※指定したスクリプトについては、後で設定します。
- ⑤ GRACE 値(リトライ回数に達するまでの監視時間)。
= ここでは **86400** 秒(1 日)を設定します。
- ⑥ リトライ回数(GRACE 値内でリトライすることができる上限回数)
= ここでは **3** 回に設定します。
- ⑦ リトライオーバーアクション(リトライ回数を超えた場合の処理)
= ここでは **continue** を指定します。



④には、以下のように rc スクリプトを指定することもできます。

例:`/etc/init.d/crond start`

プロセスの消滅を検出した場合に、何も行わない場合は、`"-"` を記述します。

4. 記述した後は、ファイルを保存して終了します。

- 再起動スクリプトを作成しましょう

再起動スクリプトは pfile に指定され、監視対象プロセスの消滅を検出するとプロセスの再起動を実行します。再起動スクリプトは通常のシェルスクリプトと同じ形式です。ここでは以下のように記述します。

1. vi で新規ファイルを作成します。

※pfile で指定したスクリプト名で作成します。

(ここでは /var/opt/HA/PS/conf/bin/restart_crond.sh とします。)

```
# vi /var/opt/HA/PS/conf/bin/restart_crond.sh
```

1

開いたファイルは次のように編集してください。

再起動スクリプト < /var/opt/HA/PS/conf/bin/restart_crond.sh >

```
#!/bin/sh
/etc/init.d/crond stop      →①
/bin/sleep 5
/etc/init.d/crond start    →②
/bin/sleep 5
exit 0                     →③
```

- ① まずプロセスの停止処理を行います。



(プロセスの消滅を検出した場合、プロセスがいなくても関連するファイルの削除等の後処理を実行する場合があるため。)

また、念のため、後処理が完了するまでの時間 sleep で待ち合わせます。

- ② プロセスを起動します。停止処理同様、プロセスの開始処理が完了するまでの時間 sleep で待ち合わせます。

- ③ exit 0 を記述します。



(ProcessSaver は再起動スクリプトの戻り値によって、正常に再起動できたかどうか判断をしていますので、再起動スクリプトは、必ず exit 0 で終了する必要があります)

2. 記述した後は、ファイルを保存して終了します。

3. chmod コマンドにより編集した再起動スクリプトに実行権をつけます。



```
# chmod +x /var/opt/HA/PS/conf/bin/restart_crond.sh
```

3

実行権がない場合、再起動スクリプトが実行されないため、必ず実行権をつけます。

以上で、ファイルの準備は完了です。

3.3. プロセス監視の開始

ProcessSaver で監視を開始しましょう。

pcheck コマンドでプロセス監視を開始します。

1. pcheck コマンドにより、監視をはじめます。

```
# /opt/HA/PS/bin/pcheck -f /var/opt/HA/PS/conf/bin/pfile_crond &
```

1



pcheck は監視を開始すると終了されるまで、フォアグラウンドにプロンプトが返ってこないため、「&」をつけてバックグラウンドでコマンドを実行します。

2. ps(1) コマンドにより pcheck が起動していることを確認します。

```
# ps -ef | grep pcheck
```

```
root  7750    7524  0   13:31:32  pts/tc    0:00   /opt/HA/PS/bin/pcheck -f  
/var/opt/HA/PS/conf/bin/pfile_crond
```

2

3.4. 監視状態の確認

ProcessSaver の監視状態を確認しましょう。

padmin コマンドでプロセスの監視状態を確認します。

1. **padmin -l** でプロセス監視(pcheck)の一覧を表示することができます。

```
# /opt/HA/PS/bin/padmin -l
ipckey   = 0x1f000501
mypid    = 7750
myname   = pcheck
pfile    = /var/opt/HA/PS/conf/bin/pfile_crond
message  = start
```

1

2. **padmin -f <pfile> -c show param** でプロセスの監視ルールを表示することができます。

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile_crond -c show param
MSG_CHECK_INTERVAL  = 5
MONITOR_INTERVAL    = 10
MONITOR_TRY_COUNT    = 2
SHM_DUMP_FILE       = /var/opt/HA/PS/log/pcheck_dump_crond
PFILE               = /var/opt/HA/PS/conf/bin/pfile_crond
MESSAGE_BOX         = start ★プロセス監視を実行しているメッセージ
MONITOR_STOP_COUNT  = 0
FAIL_PROC_COUNT     = 0
ALL_PROC_COUNT      = 1
```

2

3. padmin -f <pfile> -c show pent でプロセスの監視状態を表示することができます。

3

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile_crond -c show pent
pname           = crond
pid             = 8033 ★何番で稼働しているプロセスを監視しているかを表示
retry_count     = 0
restart_count   = 0
proc_sts        = AVAIL ★プロセスが正常に動作しているという表示
retry_over_act  = continue
rerun_time      = -----
pent_id         = 1
monitor_sts     = on
```

3.5. プロセスの消滅検出および再起動の確認

ProcessSaver の消滅検出と再起動を確認しましょう。

監視対象プロセスを消滅させ、消滅を検出して再起動されることを確認します。

※監視対象プロセスの消滅には kill コマンドを使用します。

1. ps(1)コマンドにより監視対象プロセス crond プロセスが起動していることを確認します。

```
# ps -ef | grep crond  
root 8033 1 0 13:33:27 pts/tc 0:00 crond
```

※PID および STIME を確認します。

2. 手順 1 で確認した PID を指定して、crond を強制的に停止します。

```
# kill 8033
```

3. 数秒後、自動的に以下のメッセージが表示されます。

```
# crond を停止中: [失敗]  
crond を起動中: [ OK ]
```


4. 約 30 秒後(*1)、ps(1)コマンドで crond が再起動されていることを確認します。
(手順 1 で確認した PID、STIME と異なる値であることを確認します。)

```
# ps -ef | grep crond
```

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
root	8107	1	0	13:34:50	pts/tc	0:00	crond

(*1) ProcessSaver は pfile の PARAM に設定されている MONITOR_INTERVAL 間隔でプロセスの存在をチェックしています。

今回の設定では 10 秒ごとにプロセスの存在をチェックしているため、前回チェック時から 10 秒後にプロセスの消滅を検出することになります。
また、再起動スクリプト内でも sleep によって処理が正常に終了するまでの時間を確保しているため、プロセスの消滅検出から再起動までには、今回の場合は最大で 30 秒の時間がかかります。

5. syslog ファイルにてプロセスの消滅→再起動が正常に行われたことを確認します。

```
# view /var/log/messages
```

```
:           :           :  
Jan 18 13:34:43 xxx pcheck[7750]: Process [crond, pid=8033] Down ★1  
Jan 18 13:34:43 xxx pcheck[7750]: Try to restart (crond) ★2  
Jan 18 13:34:43 xxx crond: crond 停止 failed ★3  
Jan 18 13:34:48 xxx crond: crond 起動 succeeded ★4  
Jan 18 13:35:04 xxx pcheck[7750]: (crond ,pid=8107) Up ★5  
:           :           :
```

- ★1 crond の消滅を検出したメッセージ
- ★2 crond の再起動を開始したメッセージ
- ★3 crond が停止したことを示すメッセージ
- ★4 crond が起動したことを示すメッセージ
- ★5 crond の監視を開始したメッセージ



pcheck[7750] の"7750"は pcheck の PID です。

3.6. リトライオーバーの確認

障害によっては、プロセスの消滅を検出した後再起動することができない場合があります。
リトライオーバーの状態を確認しておきましょう。

プロセスの消滅と再起動を繰り返し、リトライオーバーの状況を確認します。

1. 以下の手順を 3 回繰り返し行います。

1-1.cron d が起動していることを確認します。

```
# ps -ef | grep crond  
root 8107 1 0 13:34:50 pts/tc 0:00 crond
```

1-1

※PID および STIME を確認します。

1-2.上記手順で確認した PID を参照し、crond を強制的に停止します。

停止後、30 秒程度待ち合わせてから、処理を繰り返します。

```
# kill 8107
```

1-2

Check!!



GRACE 値に指定された時間内に pfile に設定したリトライ回数(3 回)を超えた場合にリトライオーバーとなり監視は終了します。

次回リトライまでに GRACE 値を超えた場合、リトライ回数はクリアされます。

2. crond が再起動されていないことを確認します。

```
# ps -ef | grep crond
```

2

※表示されないことを確認します。

3. padmin コマンドによりプロセスがリトライオーバーとなったことを確認できます。

3

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile_crond -c show pent
pname           = crond
pid             = 8482
retry_count     = 0
restart_count   = 3 ★リトライ回数
proc_sts        = RETRY_OVER ★リトライオーバー時の表示
retry_over_act  = continue
rerun_time      = Thu Jan 18 13:34:43 2007
pent_id         = 1
monitor_sts     = on
```

4. syslog ファイルに以下のメッセージが出力されることを確認します。

4

```
# view /var/log/messages
:           :           :
Jan 18 13:39:25 xxx pcheck[7750]: (crond) RETRY_OVER:CONTINUE ★
:           :           :
```

★ リトライオーバー時のメッセージ



リトライオーバーとなった場合の処理は、リトライオーバーアクションの指定によって異なります。今回の場合は、continue を指定しているため、監視を終了します。

リトライオーバーアクションには以下を指定することができます。

continue	: 最大リトライ回数に達したプロセスを監視対象からはずします。
exit	: クラスター環境の設定により、サーバー切り替えにつながる終了処理を行います。
Shutdown	: pcheck コマンドを終了します。

3.7. プロセス監視の復旧

3.6 章でリトライオーバーとなった場合、プロセス監視は終了します。
再度監視を始めるには以下の手順を行います。

padmin コマンドでプロセス監視を復旧します。

1. まず crond を起動します。

```
# /etc/init.d/crond start
```

1

2. 数秒後、自動的に以下のメッセージが表示されます。

```
#crond を起動中:
```

```
[ OK ]
```

2

3. ps(1)コマンドにより、crond が起動したことを確認します。

```
# ps -ef | grep crond
```

```
root 8888      1  0   13:41:38   pts/tc    0:00   crond
```

3

4. リトライオーバーとなったプロセスの監視を以下のコマンドにより再開することができます。

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile_crond -c reload
```

4

5. プロセス監視が再開されたことを確認します。

5

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile_crond -c show pent
pname           = crond
pid             = 8888
retry_count     = 0
restart_count   = 0
proc_sts        = AVAIL ★プロセスが正常に動作しているという表示
retry_over_act  = continue
rerun_time      = -----
pent_id         = 1
monitor_sts     = on
```

6. syslog ファイルでプロセス監視が再開されたことを確認します。

6

```
# view /var/ log/messages
:
:
Jan 18 13:40:06 xxx crond: crond 起動 succeeded ★1
Jan 18 13:41:30 xxx padmin[8821]: Command execute. [reload] pfile=/var/opt
/HA/PS/conf/bin/pfile_crond ★2
Jan 18 13:41:52 xxx pcheck[7750]: (crond ,pid=8888) Up ★3
:
:
```

- ★1 crond が起動したことを示すメッセージ
- ★2 padmin がコマンドを実行したメッセージ
- ★3 crond の監視を開始したメッセージ

3.8. プロセス監視の終了

ProcessSaver のプロセス監視の終了手順を確認しましょう。

padmin コマンドでプロセス監視を終了します。

1. 以下のコマンドにより、プロセス監視を終了することができます。

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile_crond -c  
shutdown
```

1

2. ps(1)コマンドにより、pcheck が起動していないことを確認します。

```
# ps -ef | grep pcheck
```

2

表示されないことを確認。

3. padmin コマンドによりプロセス監視が終了したことを確認します。

```
# /opt/HA/PS/bin/padmin -l
```

3

表示されないことを確認。

4. syslog ファイルに以下のメッセージが出力されることを確認します。

```
# view /var/log/messages  
:  
:  
Jan 18 13:47:43 xxx pcheck[7750]: Shutdown pcheck ★1  
Jan 18 13:47:48 xxx padmin[9501]: Command execute. [shutdown] pfile=/var  
/opt/HA/PS/conf/bin/pfile_crond ★2  
:  
:
```

4

★1 pcheck を終了したメッセージ

★2 padmin がコマンドを実行したメッセージ

4. おわりに

以上、ProcessSaver の基本的な動作となります。

その他の監視対象プロセスについても、同様に設定・監視を実行してみましょう。

また、この他にも同一名プロセスの監視やグループ監視などシステムにあわせた多種多様な監視を行うことができます。

高度な監視の方式については、「CLUSTERPRO MC ProcessSaver 2.2 for Linux ユーザーズガイド」をご覧ください。

CLUSTERPRO
MC ProcessSaver 2.2 for Linux
はじめての ProcessSaver

2017 年 4 月 第 3 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 2017

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