

Printer Daemon 説明書

株式会社 アックス

2004 年 2 月 23 日版

1 はじめに

CUPS では、プリンタのステータスは、プリント出力時にしか得る事ができない。そのため、Bidi を用いたとしても、プリント出力せずにインク残量等を知る事ができない。そこで、CUPS のデーモンに加えて、常時動作しているデーモンを設け、Printer Status Utility からのプリンタのステータス取得要求に答えられるようにする。これを Printer Daemon と呼ぶ。

デバイスの排他制御、プリンタのコマンド体系等のためプリンタのステータス取得は、プリント出力と無関係に行なう事ができない。そのため Printer Daemon は、ステータス取得だけでなく、プリント出力をも制御する。CUPS においては、プリント出力は backend が担うべきものであり、CUPS としては、非常に変則的であるが既存の CUPS の体系を大きく変えずに実現するために、やむを得ずこのようになっている。

プリンタデーモンは、Spooler 機能、他のホストのプリンタとの連携機能も持つ。

2 概要

Printer Daemon は、図 1 のように 4 種類のプロセスから構成される。

各プロセスは、以下の機能を持つ。

- リスナ

リスナは、常時動作し、CUPS のバックエンドからのプリント出力、Printer Status Utility Type B からのプリンタステータス取得の要求のためのソケットの受付を行なう。ソケットの回線を確立すると、要求のポート番号に従って、スプール・プロセスまたはステータス・プロセスを起動する。このとき確立した回線は、各プロセスに受けつがれて、各プロトコルの処理が任される。

- スプール

backend からプリント出力データを受けとりスプールファイルに書き出す。ひとつのジョブデータを処理が終了すると、このプロセスも終了する。

- ステータス

Printer Status Utility からの要求を処理する。必要に応じて、アウトプット・プロセスに接続しプリンタからステータスを取得する。Printer Status Utility からの回線が存在する限りこのプロセスも存在する。複数の Printer Status Utility からひとつのプリンタのステータス取得の要求がある場合は、このプロセスもひとつのプリンタにたいして複数起動される。

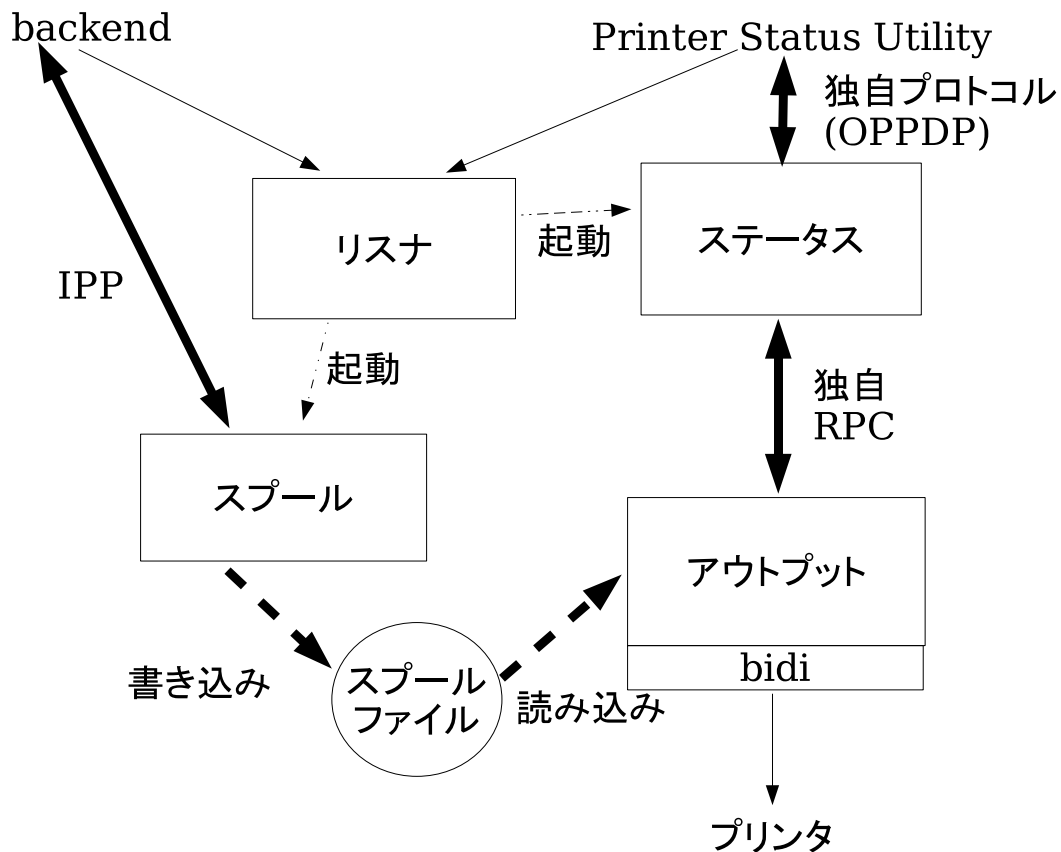


図 1: プリンタデーモンの構成

- アウトプット

プリンタと直接やりとりを行なうプロセス。bidi を利用して、必要に応じてプリント出力、ステータス取得を行なう。プリンタに対して、ただひとつ起動され、常時動作している。担当のプリンタに対するスプールファイルが存在を検知すると、プリント出力を開始する。

使用しているプロトコルは、以下の通りである。

- backend とリスナ・プロセス間

IPP を利用する。従って、プリント出力には IPP backend を利用する。ただし、現在はプリント出力に必要な部分のみ実装されている。

- Printer Status Utility とステータス・プロセス間

HTTP の上に、独自のプロトコルを実装している。ここでは、仮に OPPDG と呼ぶことにする。詳細は、後述する。

- ステータス・プロセスとアウトプットプロセス間

独自の RPC プロトコルを実装している。プロトコルは、Vector API IP Wrapper で利用しているものと同じである。プロトコルの詳細は、IP Wrapper のドキュメントを参照の事。

2.1 動作概要

ここではプリンタ・デーモンの動作の概要を述べる。

2.1.1 起動時

1. システム起動時には、リスナ・プロセスを起動する。
2. リスナ・プロセスは、設定ファイルにある自身が管理すべきプリンタの分だけアウトプット・プロセスを起動する。
3. その後、プリント出力用のソケット (ポート番号 22222) およびステータス取得用のソケット (ポート番号 22223 への接続要求を待つ。
4. アウトプット・プロセスは、ステータス・プロセスからの接続要求を待ちながら、スプールファイルができるのを待つ。

2.1.2 プリント出力時

プリント出力要求時には以下のように動作する。

1. backend がリスナ・プロセスの待つソケットに接続要求を出す。
2. リスナ・プロセスは、接続を受け付け、TCP 回線を確立後スプール・プロセスを起動する。
3. スプール・プロセスは、IPP を処理し、プリント・データをスプール・ファイルに書き込み、終了する。
4. アウトプット・プロセスは、スプール・ファイルを検出すると、プリンタ・デバイスをオープンし bidi を初期化する。プリンタ・デバイスおよび bidi は、一度起動するとエラーが生じない限り、そのまま使い続ける。
5. スプール・ファイルのデータを bidiWrite を使って、プリンタに出力する。
6. 出力が終了すると、スプール・ファイルを削除する。

2.1.3 ステータス取得時

プリンタ・ステータスの取得を行なう場合は、以下のように動作する。

1. Printer Status Utility がリスナ・プロセスの待つソケットに接続要求を出す。
2. リスナ・プロセスは、接続を受け付け、TCP 回線を確立後ステータス・プロセスを起動する。
3. ステータス・プロセスは、OPPD 処理し、アウトプット・プロセスの待つソケットに接続要求を出す。
4. アウトプット・プロセスは、接続要求を受け付け、RPC の処理を開始する。
5. ステータス・プロセスは、RPC によりアウトプット・プロセスにステータス取得要求を出す。

6. アウトプット・プロセスは、bidi に対しステータス取得要求を出し、得られたステータスをステータス・プロセスに返す。
7. ステータス・プロセスは、得られたステータスを Printer Status Utility に返す。

3 設定ファイル

プリンタデーモンの設定ファイルは、管理するプリンタを記述し、

/etc/oppd.conf

とする。

形式は、以下の通り。

1 行につきプリンタひとつを以下の形式で記述する。

<プリンタ名><TAB><デバイス名><TAB><BIDI 名><TAB><コメント>

<TAB> は、ひとつのタブコードを示します。

<デバイス名> は、プリンタに繋がるデバイスファイル名を / で始まるフルパスで記述する。<デバイス名> が / で開始されていない場合はリモートプリンタで、URI を記述する。ただし、リモートプリンタへの出力はサポートされていない。リモートプリンタは他のプリンタデーモンが管理するプリンタのステータスの取得のみがサポートされている。その場合は、以下の形式の URI を記述する。

oppd://<ホスト名>:<ポート番号>/<プリンタ名>

<BIDI 名> は BIDI のモジュール名を、記述する。リモートプリンタの場合は、記述しない。# で始まる行はコメント行である。

4 OPPDP

OPPDP は、プリンタデーモンと Printer Status Utility Type B(以後、PSUB と記す) 間のプロトコルである。ここでは、このプロトコルについて説明する。

4.1 概略

通信は、HTTP/1.1 の上に構築する。プリンタデーモンがサーバ側、PSU B がクライアント側となる。プリンタデーモンのプロセスのオーバーヘッドを避けるため、回線は基本的にはパーシステントとし、HTTP のヘッダ "Connection: close" は指定しない。

このプロトコルでは、以下の三つのコマンドを定義する。

1. プリンタ・リスト取得
2. プリンタ・ステータス取得
3. プリンタ制御

次項では、これらについて説明する。

4.2 各コマンドの詳細

4.2.1 プリンタ・リスト取得

プリンタ・デーモンがサポートしているプリンタのリストを取得する。

メソッドに、GET を使用し、要求 URI として */list* を指定する。プリンタのリストは、レスポンスの body に返る。リストの形式は、各行が、ひとつのプリンタを表し、以下の形式となる。

<printername> <comment> CRLF

プリンタ名には、空白文字を含まない。printername と comment の間は、空白 (SP コード) で区切られ、以後 CRLF まで comment とみなされる。

4.2.2 プリンタ・ステータス取得

プリンタのステータスを取得する。

メソッドに GET を使用し、要求 URI として、*/status/ <printer 名>* を指定する。ここで、<printer 名> は、プリンタ・リスト取得で得られたプリンタ名のうちのひとつを指定する。

読み込みモードとロケール指定文字列は、以下の形式で URI に引数として <printer 名> の後ろに指定する。

idReadMode=<readmode>&LANG=<lang>

ここで <readmode> は、読み込みモードを数字で指定したもの、<lang> は、ロケール指定文字列

これらの引数は、省略可能で省略時には、読み込みモードは 1 となり、LANG は C となる。

ステータスは、レスポンスの body に返る。その形式は、bidi で決められたものとする。

4.2.3 プリンタ制御

プリンタへ制御コマンドを送る。

メソッドに POST を指定し、要求 URI として、*/status/ <printer 名>?request =< idRequest >* を指定する。ここで、<printer 名> は、プリンタ・リスト取得で得られたプリンタ名のうちのひとつを指定する。<idRequest> は、要求 id を指定し、必須である。

制御コマンドは、要求の body に指定し、その形式は bidi で決められたものとする。Length フィールドは必須である。

4.3 エラーレスポンス

エラーレスポンスとして、以下のものが有りうる。

- 301 Moved Permanently

指定したプリンタが他のホスト上にあった場合これで Redirect する可能性がある。

- 400 Bad Request

HTTP 形式上おかしいリクエスト

- 404 Not Found

規定にないパスを指定されたか、存在しないプリンタ名を指定した。

- 405 Method Not Allowed

PUT など、現在利用していないメソッドを指定した。

- 408 Request Timeout

タイムアウトした。

- 411 Length Required

Length フィールドが必須なのに存在しない。

- 413 Request Entity Too Large

内容が大き過ぎる。

- 414 Request-URI Too Long

URI が長過ぎる。

- 501 Not Implemented

指定の機能は、実装されていない。

- 503 Service Unavailable

サーバが機能状態にない。

- 505 HTTP Version Not Supported

HTTP バージョンがおかしい。

- 500 Internal Server Error

サーバ内部のエラーの他、BIDI 等でのエラー

Body には、以下の形式で詳細のエラー番号が入る。

`ecode:<詳細エラー番号>`

詳細エラー番号は、以下のとおり。

- -1 から-99 BIDI の返すエラー番号
- -100 メモリ不足
- -110 設定ファイルのオープンに失敗
- -111 設定ファイル読み込みエラー

4.4 リモート・プリンタ

プリンタデーモンにステータスの取得や Ctrl を書き込みを行ったとき、指定のプリンタがリモートプリンタであった場合、

301 Moved Permanently

をレスポンスで返す。

このとき、Location フィールドの値として、リモートプリンタの URI を返す。この URI は、次の形式とする。

oppd://<ホスト名>[:<ポート番号>]/<プリンタ名>

これを受け取ったら、PSUB 側では、この URI が表すホスト名ポート番号に新たなプリンタ名で接続を仕直す。このフィールドの URI には、status などのコマンドおよび idReadMode 等のパラメータは、つかないので必要に応じて付加する。