

## 教育用計算機システムにおけるプリンタシステムに 求められる要求とその実装

梶 田 秀 夫<sup>†1</sup> 小 川 剛 史<sup>†1</sup> 齊 藤 明 紀<sup>†2</sup>  
中 村 匡 秀<sup>†3</sup> 近 藤 弘 一<sup>†4</sup> 中 西 通 雄<sup>†5</sup>

教育用計算機システムでは、運用コスト削減が大きな課題となっている。本論文では、特にプリンタシステムに着目し、無駄な印刷を減らすことを目的とした。そのために、教育環境における一般的なプリンタシステムへの要求事項を、ユーザ側の視点と管理者側の視点の両面から、(1) 事前確認機能、(2) 取捨選択機能、(3) 総印刷枚数制限機能、(4) プリンタアクセス制限機能、の4点に分けて整理する。その上で、我々の教育用計算機システム上に実装を行った。本プリンタシステムは、印刷枚数制限機能を持たせた LPRng ベースのプリンタサーバ上と、Linux クライアント上の GUI ツールから構成されている。さらに、サーバでは、Postscript エンジンと通信する手法と PrinterMIB を用いる手法の2方式を選択できる。また、このプリンタシステムを稼働させたことによって、総印刷枚数が削減されたことを確認した。

## Requirements and Implementation of a Printing Subsystem for Educational Computer System

HIDEO MASUDA,<sup>†1</sup> TAKEFUMI OGAWA,<sup>†1</sup> AKINORI SAITOH,<sup>†2</sup>  
MASAHIDE NAKAMURA,<sup>†3</sup> KOUICHI KONDOU<sup>†4</sup>  
and MICHIO NAKANISHI<sup>†5</sup>

This paper presents an effective framework to avoid waste printing in an educational computer system. We first point out primary four requirements from both administrator and user sides, (1) preview, (2) shrink, (3) quota and (4) authorize function. Then, to meet the requirements, we have implemented a printing subsystem in our educational computer system. The subsystem consists of LPRng-based printer server and GUI applications for Linux clients. The server uses either "raw socket mode" or "printerMIB" for obtaining the number of printings from printers, and manages printing quota. Statistics show that the number of papers printed is significantly reduced after deploying the proposed subsystem.

### 1. はじめに

教育の情報化が進むにつれ、教育用計算機システムを利用する授業も増えてきており、教育用計算機シ

ステムの運用管理の効率化やコストの削減も急務となってきた<sup>1)-3)</sup>。システムの運用コストには、計算機自体の日常管理もさることながら、規模が大きくなるにつれ、消耗品の手当てなどのランニングコストの問題も大きくなっていく。本論文では、特にプリンタに関するランニングコストに着目する。

プリンタシステムのランニングコストの削減には、無駄な印刷を減らし、限られた資源(トナー、紙)を有効に活用させることが重要である。しかし、多くの教育用計算機システムでは、一般利用者に直接課金するといった仕組みはあまりとられていないため、利用者のコスト意識は低くなりがちであり、不用意な印刷指示やコピー機代わりの大量印刷など、無駄な印刷を行う姿がよく見られる。特に対象とする利用者の規模が拡大するにともない、教育やモラルだけではコスト意識

<sup>†1</sup> 大阪大学サイバーメディアセンター  
Cybermedia Center, Osaka University

<sup>†2</sup> 鳥取環境大学情報システム学科  
Department of Information System, Tottori University  
of Environmental Studies

<sup>†3</sup> 奈良先端科学技術大学院大学情報科学研究科  
Graduate School of Information Science, Nara Institute  
of Science and Technology

<sup>†4</sup> 同志社大学工学部  
Department of Electrical Engineering, Doshisha University

<sup>†5</sup> 大阪工業大学情報科学部  
Faculty of Information Science and Technology, Osaka  
Institute of Technology

を高めることは困難になるため、無駄な印刷を抑制する技術的な仕組みが必要となってくる。このためには、

- (1) 利用者が印刷枚数を減らしやすいようにすること、
- (2) 管理者がきちんと出力枚数を管理できること、が必要である。さらに、利用者にも無駄な出力を減らすように教育するとともに、節約するための操作しやすいインタフェースを提供することが必要である。

本論文では、まず 2 章でプリンタシステムに求められる要求分析について述べる。3 章で従来研究や既存の製品について述べる。4 章で UNIX サーバ上で稼働する印刷枚数制限機能の構成について述べる。5 章では Linux PC 上で稼働する印刷用 GUI ツールの実装について述べる。さらに、このシステムを実際に運用したので、印刷コスト削減の効果について 6 章で考察する。

## 2. 要求分析と機能要件

まず本論文では、プリンタシステムのコストに関して、以下を前提とする。

- プリンタの利用課金は、利用者に直接課金しない。
- プリンタの消耗品の経費負担方法が複数存在する。
- システム導入コストは、できるのであれば抑えたい。

この前提のもとで、教育環境におけるプリンタシステムに要求される機能は、以下の 4 点に分けて整理できる。

- (1) 事前確認機能 (preview)  
利用者に印刷イメージを確認してから印刷する習慣を身につけさせるために、どのように印刷されるかを、利用者が容易にプレビューできる必要がある。これにより「とりあえず出力してから確認しよう」という行動傾向を減らし、無駄な印刷の削減が期待できる。
- (2) 取捨選択機能 (shrink)  
利用者が、印刷物のうち本当に必要なページを取捨選択して印刷ができる必要がある。また、複数ページを縮小して 1 枚に収めるといった機能も必要である。これにより、不必要な印刷を減らすことが期待できる。
- (3) 総印刷枚数制限機能 (quota)  
総印刷枚数に制約があると分かれば、利用者のコスト意識を高めることが可能となる。また、所属学部ごとの利用傾向やプリンタ設置場所ごとの消耗品の経費負担方法の違いなどから、単一の制限ではなく、利用者ごとや利用者の所属グループごとの制限、プリンタ設置場所ごとの

制限ができることが求められる。

- (4) プリンタアクセス制限機能 (authorize)  
枚数制限機能があっても、印刷枚数をきちんと取得する仕組みが正しく稼働する必要がある。制約に対する抜け道がないように工夫することが求められる。

これらの機能を実現するにあたり、管理者側の立場として求められる要件には、以下があげられる。

- A1: 印刷枚数制限の仕組みは、できるだけ集中管理できること。
- A2: 特殊で高価なプリンタでなくても、適切に動作すること。
- A3: 印刷枚数制限の仕組みを迂回して印刷できないこと。
- A4: 利用者ごとの印刷枚数上限を設定できること。
- A5: 利用者ごとや利用者の所属グループごと、プリンタ設置場所ごとに割当て量を変えるといった柔軟な管理が可能であること。

また、利用者側の立場として求められる要件は、以下があげられる。

- U1: すべての印刷出力を同じインタフェースでプレビューできること。
- U2: 印刷するページを選択したり、複数ページを縮小してまとめて印刷したりできること。
- U3: 自分の出力枚数の累計や残り枚数を知ることができること。
- U4: プリンタへの出力状態や待ち行列を知ることができ、また、出力するプリンタを容易に選択できること。

## 3. 既存の研究や製品との違い

従来の多くのネットワーク対応プリンタでは、以下のような機能が欠けている。

- F1: 印刷アカウント処理機能
- F2: ネットワークからのアクセスに対するサービス制限機能  
たとえば、ftp による印刷のみを禁止したり、特定のホストからのみ lpd の要求を受け入れる、など。このため、単体では教育環境におけるプリンタシステムの要件を満たしていない場合が多い。

商用の製品では、文献 8) のプリンタシステムや文献 9) のプリンタシステムが存在しており、これらの機能が含まれているが、コストが高く、専用のプリンタのみに対応していることから、要件 A2 を満たすことができない。また、要件 A5 を満たすようにはできず、多くの機能を追加実装する必要がある。

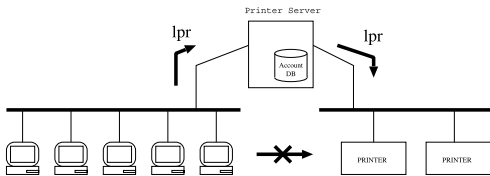


図 1 印刷プリンタシステムの構成

Fig.1 Configuration of printing subsystem.

また、文献 2) のシステムでは、印刷ジョブを Postscript としたうえで、Postscript 解釈プログラムとしてフリーの実装である Ghostscript を利用し、

- (1) プリンタに印刷ジョブを送る前に、プリンタサーバ上で動作する Postscript 解釈プログラムを用いて、その印刷ジョブを評価し、印刷枚数を求める、
- (2) 所定の総印刷枚数を超えてしまうのであれば印刷ジョブを破棄する、
- (3) プリンタに印刷ジョブを送る、
- (4) 印刷枚数を記録する、

という方式を提案している。この方式では、要件 A2 は解決できているが、Ghostscript と、プリンタに搭載されている Adobe 社の Postscript 解釈プログラムでは、仕様が完全には一致していないため、この 2 つの解釈プログラムの結果は異なることがある (Ghostscript では印刷ジョブの解釈自体が失敗してしまう、ということが生じる)。つまり、印刷枚数を正しくカウントできない場合が存在してしまう。

また、印刷形式を操作する Windows 用の GUI は、Windows 標準の仕組みを含め、いろいろ存在しているが、Linux などの UNIX 系 OS ではあまり充実していない。

#### 4. プリンタシステムの実装

本章では、印刷ジョブを Postscript とした場合の実装について述べる。

##### 4.1 システム構成

プリンタの設置場所の自由度を上げるために、ネットワークに直接接続可能なプリンタが広く使われており、本システムでも、ネットワーク対応の Postscript プリンタを前提とした。そのうえで、要件 A1, A2, A3 を実現するために、図 1 に示す構成で設計した。

本構成では、1 台の UNIX ワークステーションをプリンタサーバとして設置し、すべての印刷ジョブをこのプリンタサーバにいったんまとめ、プリンタサーバで印刷の可否をチェックした後、印刷可能の場合のみ本来のプリンタ本体に印刷ジョブを送る。したがって、プリンタ自体に機能 F1 がなくても印刷枚数を集中し

て管理できる。また、機能 F2 に関しては、クライアントとなる Linux PC 群とプリンタ群を異なるネットワークに配置し、両群の間をプリンタサーバで通信するようにシステムを構成した。これにより、プリンタサーバで、印刷ジョブ以外の通信をすべて遮断する (IP forwarding を無効にするなど) ことで、クライアントから直接プリンタに印刷ジョブを送ることは不可能となる。さらに、プリンタはプリンタサーバと同一のブロードキャストドメイン内に配置し、プリンタサーバ以外とは通信できないように、default gateway などを設定しなければ、より安全となる。

ただし、この構成だけでは、プリンタのネットワークポートや LPT ポートに直接ケーブルを接続するといった、物理的な迂回手段は防げない。これに対しては、接続ポートを物理的に繋ぎ替えられなくすることで防げるが、プリンタ本体のメンテナンスの都合上一般的には難しい。接続ポートを隠すもしくは容易には手が届きにくいようにプリンタを設置するようにすることで、ある程度抑止できると考えられる。

##### 4.2 印刷枚数のカウント

要求 A4 を実現するために、印刷枚数をカウントする機構が必要となる。Linux を含む多くの UNIX 系 OS では、印刷物のフォーマットとして、Postscript が事実上の標準となっている。しかし、一般に Postscript フォーマットは、印刷物が最終的に何ページになるかを正確にカウントすることは困難である。なぜなら、Postscript はページ記述言語であり、繰返しなどの制御構造も記述できるためプログラムを解釈しなければ、印刷ページ数が分からないためである。

そこで本論文では、プリンタサーバがプリンタに印刷枚数を問い合わせる方法を採用する。このアルゴリズムは以下ようになる。

Step1: ジョブをプリンタに送付する前にプリンタの総印刷枚数カウンタ値  $total$  を取得する。

Step2: ジョブをプリンタに送付する。

Step3: プリンタが印刷を完了したかどうかをポーリングで確認する。

Step4: 印刷完了後に、総印刷枚数カウンタ  $total'$  を取得する。

Step5:  $total' - total$  を、このジョブの印刷枚数として記録する。

また、印刷要求は、LPD プロトコル (Line Printer Daemon Protocol, RFC1179) によってプリンタやプリンタサーバ間でやりとりすることが普通であり、lpd という実装がよく知られている。プリンタに対して総印刷枚数を問い合わせるため、印刷ジョブを送る前後

に、別途プリンタと直接通信を行う必要があるが、通常の lpd は、印刷ジョブの転送までしかできない。また、印刷ジョブを受け付けるホストの制限程度の機能しかなく、利用者ごとやプリンタの設置場所ごとの制約をつけにくい、という問題もある。そこで、LPRng (lpr Next Generation) という実装を利用することとした。

最近、LPRng よりも汎用性が高い CUPS (Common Unix Printint System) も使われるようになってきているが、本論文で紹介する実装方法にも問題なく適用できる。

#### 4.3 総印刷枚数の取得方法

4.2 節のとおり、印刷ジョブに対する印刷枚数を取得するために、プリンタから印刷枚数情報を取得する必要がある。

このためには、プリンタとなんらかの双方向通信を行って情報を取得しなければならない。しかし、LPD プロトコルは基本的に単方向通信であり、エラーコード程度しか返せない。この情報取得には、多くのネットワーク接続プリンタの仕様から、AppleTalk、セントロニクス、PrinterMIB (RFC1759/RFC3805)<sup>6),7)</sup> を用いた SNMP などが考えられる。AppleTalk は、IP とは異なる別の通信プロトコルを利用しなければならない。またセントロニクスは、設置場所の自由度が高いという、せっきくのネットワーク接続プリンタの利点が失われてしまう。PrinterMIB は、きちんと実装されているプリンタは広まってきているが、世代の古いプリンタでは、正しく実装されていないプリンタも存在している。

したがって、本論文では、

- (1) PrinterMIB が使えない場合に対応する、'raw socket mode' と呼ばれる方法
  - (2) PrinterMIB を直接利用する方法
- の両方を実装した<sup>4),5)</sup>。

##### 4.3.1 'raw socket mode' による実装

'raw socket mode' は、プリンタの PostScript エンジンと TCP を用いて通信する仕組みで、HP 社製の Postscript プリンタなど、比較的多くのプリンタに実装されている。この仕組みを利用することにより、プリンタ内部の Postscript エンジンに対して、

```
statusdict begin pagecount (*) print =
flush end
```

というコマンドを送ると、

\*数値

という形で、プリンタ内部の総印刷枚数カウンタ値が取得できる。

しかし一般に、プリンタは、ラスタライザ (Postscript エンジン) とメディアパス (カウンタを含む紙が通る部分) がパイプライン的に動作しており、Postscript エンジンが印刷完了になったとしても、総印刷枚数カウンタが正しい値になっているとは限らないことが分かった。したがって、実際の印刷が完了していないとき、本方法で総印刷枚数を問い合わせると、最後の印刷が完了していないため、少ない枚数を返してしまうことが起こりうる。

そこで、総印刷枚数の問合せステップを、以下のようアルゴリズムで実装した。

Step3-1: 1 枚目が出力されるのにかかると予想される時間  $t_1$  だけ待つ。 $t_1$  は、プリンタのファーストプリント時間性能に対して少し余裕を持たせた値とする。具体的な値については、6.1 節で議論する。

Step3-2: 総印刷枚数カウンタ取得コマンドを送って、カウンタ値を読み取る。

Step3-3: 1 枚が排出されるのにかかると予想される時間  $t_2$  だけ待つ。 $t_2$  は、プリンタの印刷性能に対して少し余裕を持たせた値とする。具体的な値については、6.1 節で議論する。

Step3-4: 総印刷枚数カウンタ取得コマンドを再度送って、前回のカウンタ値から変化があった場合は印刷中と判断して、Step3-3 に戻る。

Step4: 印刷完了として、前回取得したカウンタ値を返す。

さらに、プリンタの機種によっては、コマンド送信時の情報ではなく、TCP セッションが確立したタイミングの情報を返すものもあることが分かった。したがって、実装にあたっては、取得コマンドを送信するたびに TCP セッションを張り直すようにしなければならない。

また制限事項として、プリント途中でのプリンタ故障や Postscript エンジンが長時間ビジー状態の場合は、Step3-2 や Step3-4 のカウンタ取得コマンドが送付できなくなりエラーになる。このときは、印刷ジョブ自体がキャンセルされたと判断している。プリント途中での紙切れの場合は、出力されたところまでを計測することになる。この場合も、紙補給時には必ずプリンタをリセットする、という運用をすることにより、

本学のシステムで導入されている EPSON 社製 LP-8400PS3 は、無意味な値を返していることが分かった。

EPSON LP8400PS3 は前者だが、FujiXEROX Docuprint C2221 は後者であった。

大きな問題にはならない。

#### 4.3.2 PrinterMIB による実装

PrinterMIB では、プリンタ内部の各種情報を、MIB と呼ばれる単位で処理している。本システムの目的に利用可能な MIB を調査したところ、総印刷枚数カウンタ取得には、以下のいずれかの値が利用可能であると判断できた。

- PrtMarkerLifeCount (mib-2.43.10.2.1.4)  
プリンタが設置されてからの総印刷枚数が記録されている。
- PrtMarkerPowerOnCount (mib-2.43.10.2.1.5)  
プリンタの電源が入れられてからの総印刷枚数が記録されている。

なお、プリンタのメディアの単位に関する情報が、PrtMarkerCounterUnit (mib-2.43.10.2.1.3) で取得できる。この情報が、impression (紙の面数) もしくは sheets (紙の枚数) の場合は問題ないが、tenThousandsOfInches, feet, meters のような長さの場合は、枚数としてのカウントはできない。

また、印刷完了の確認には PrinterMIB のうち、以下のいずれかの値が利用可能であると判断できた。

- PrtConsoleOnTime (mib-2.43.13.17.6.1.2.1)  
プリンタ操作盤の LED の On/Off を表す。LED の内容は、PrtConsoleDescription (mib-2.43.13.17.6.1.5.1) で参照できるが、機械的に判定できる統一的な記述ではないので、設定時に管理者が判断してやる必要がある。
- PrtMediaPathStatus (mib-2.43.13.4.1.11.1)  
メディアパス (紙送り部分に相当) の状態を保持する。
- PrtMarkerStatus (mib-2.43.10.2.1.15)  
マーカー (ラスタライザ部分に相当) の状態を保持する。

この値は、メディアパスを紙が通過する事象と関連付けられているため、“raw socket mode” による実装にあった問題はない。

これらの値を利用することで、総印刷枚数の問合せステップを、以下のようなアルゴリズムで実装できた。

Step3-1: 時間  $t_a$  だけ待つ。この時間は、SNMP エージェントの負荷にならない程度の頻度で行えばよい。具体的な値については、6.1 節で議論する。

Step3-2: 印刷完了かどうかを問い合わせ、まだであれば、Step3-1 に戻る。

Step4: 総印刷枚数カウンタ取得コマンドを送って、カウンタ値を読み取る。

ただし、PrtMediaPathStatus と PrtMarkerStatus の変化が、PrtMarkerLifeCount のカウントアップが完了してから数秒遅れて idle 状態に戻ったとなるプリンタが存在することも分かった。そこで、通常は PrtConsoleOnTime を利用し、もし利用できないプリンタであった場合にのみ PrtMediaPathStatus や PrtMarkerStatus を利用する。その場合は、実際の印刷完了後暫くは次の印刷ジョブを送ることができないので、スルーットが低下することになる。

#### 4.4 印刷制限枚数の管理

要求 A5 を実現する為に本システムでは、プリンタサーバ上に印刷ログデータベースを構築している。本節では、印刷ログデータベースの更新と印刷制限ルールについて説明する。

##### 4.4.1 ログの解釈とデータベースの更新

印刷ログデータベースは、印刷ジョブごとのデータベースとプリンタグループごとのデータベースの2種類を持つ。

まず、プリンタに対する印刷ログには、

- 印刷を要求した利用者名
- 印刷先のプリンタ名
- 印刷ジョブの受け取り日時
- 印刷ジョブ投入前の当該プリンタの総印刷枚数
- 印刷ジョブ終了後の当該プリンタの総印刷枚数

が記録されているので、この情報を、印刷ジョブごとにまとめ、どの利用者がどのプリンタに対して何枚印刷したのか、というデータ形式のデータベースに記録する。

また、我々のシステムでは、教室や分散端末室といった部屋の単位でグループ化して端末やプリンタを管理しているため、割当て量の変更の操作も、部屋の単位で行っている。そこで、設置部屋を単位としたプリンタグループごとに、どの利用者が総計何枚印刷したか、というデータ形式のデータベースに記録する。

あるユーザの総印刷枚数が、制限値を超えているかどうかの判定には、後者のデータベースを参照することにより、高速化を図っている。前者のデータベースは、ふだんは直接利用することはないが、制限枚数を超えた利用者に対してそれぞれの印刷ジョブ情報を提示して、理由を問うたりする目的に利用されている。

プリンタによっていずれかが存在しない場合があるが、調査したプリンタには両方とも存在した。

FujiXerox Docuprint C2221 では、「オンライン」の LED、「処理中」の LED「エラー」の LED の3つが定義されていたため、処理中を示す LED を使用した。一方、RICOH NX650S では、PrtConsoleDescription の表記だけでは判読できず、それぞれの LED が何を表しているかは、実際のコンソール上のランプのつき方と、SNMP で得られる値を対応付ける調査が必要であった。

表 1 LPRGUI/LPRMGUI の機能と要求との適合性  
Table 1 Requirements and functions of LPRGUI and LPRMGUI.

要求	機能名	機能	関連コマンド
U1:preview	プレビュー	印刷イメージのプレビュー	gv
U2:shrink	ページ選択	印刷ページの選択	psselect
	n up	複数ページの縮小印刷	psnup
	印刷方向	印刷方向の変更	a2ps, ps2ps
U3:quota	プリンタ選択	印刷先のプリンタ選択	lpr, printcap
	状況表示	印刷可能な枚数の表示	lpacctinfo
	印刷不能表示	制限値超えの印刷不能表示	lpacctinfo
U4:job	印刷ジョブ情報	プリンタの状態表示	LPRMGUI, lpq
	印刷ジョブ削除	プリンタの印刷ジョブ削除	LPRMGUI, lprm
Misc.	テキスト変換	plain text 形式を Postscript 形式に変換	a2ps
	印刷物フィルタ	印刷不能形式 (音声ファイルなど) の除去	file
	多国語対応	表示の言語選択機能	環境変数 LANG

#### 4.4.2 印刷制限ルール

本システムでは、印刷枚数の制限値を、利用者の所属学部と端末室の組合せで設定できるようにした。これにより、センター経費ではなく学部経費で用紙やトナーを補充していたりするような場合や、用紙自体を個人に持ち込ませるような運用をする場合など、下記の Ex2 のような対応ができるようになる。

また、利用者ごとや端末室ごとに、管理者によって制限値を緩和させる仕組みも用意した。これは、どうしても制限値を超えるような印刷が必要になったりした場合に利用できる。我々のシステムの運用では、基本的には半年に 300 枚までという制限を設定しているが、この制限値を超えた場合でも、利用者が指導教員に追加申請書を書いてもらうことで、一度だけ 300 枚の追加を行っている。

本システムでの印刷制限ルールの例をあげる。

- Ex.1 一般利用者は、合計 300 枚が印刷制限枚数である。
- Ex.2 理学部の学生は、理学部用の部屋では 1,000 枚、他の部屋では合計 300 枚が印刷制限枚数である。
- Ex.3 管理者は、一般教室と管理用の部屋では無制限、他の部屋では、それぞれ 100 枚ずつが印刷制限枚数である。

## 5. GUI アプリケーション

本章では、Linux で稼働する、GUI 印刷アプリケーションについて説明する。

### 5.1 プリントパネル

Linux をはじめとする UNIX 系の OS では、Mi-



図 2 LPRGUI

Fig. 2 Screen shot of the LPRGUI.

crosoft Windows などの GUI ベースのシステムに比べて、初心者にも使いやすい印刷用インタフェースが欠けている。逆に Linux では、多くのフィルタプログラムを組み合わせることで高度な処理が可能であるが、初心者にはやはり使いにくいことは否めない。

そこで、LPRGUI という GUI ベースのアプリケーションを開発した (図 2)。LPRGUI は、Linux の標準コマンドの lpr に対するフロントエンドとして設計されており、lpr と置き換えて使うことができる。また、LPRGUI は、Perl/Tk で記述された 1,200 行ほどのプログラムである<sup>3)</sup>。表 1 に、LPRGUI の機能をまとめた。表中の lpacctinfo は、ユーザの印刷枚数情報を取得するヘルパーアプリケーションであり、ディスク利用量などの Quota 制限値をまとめて表示するプログラムから共通して呼び出せるように、LPRGUI から分離して作成している。

### 5.2 印刷ジョブ制御パネル

Linux では、印刷ジョブの制御に、現在の印刷状況

我々のシステムでは、利用者名から機械的に所属学部が判定できるようにしている。



図 3 LPRMGUI  
Fig. 3 Screen shot of the LPRMGUI.

を表す `lpq` と、印刷ジョブを削除する `lprm` が使われている。これらは一般に組み合わせて使われるため、これらの両方の機能を持った LPRMGUI という GUI ベースのアプリケーションを開発した (図 3)。LPRMGUI は、`lpq` と `lprm` のフロントエンドとして使えるように設計してあり、`lprm` と置き換えて使うことができる。

LPRMGUI は、選択されたプリンタの印刷ジョブ情報を表示することができる。利用者は、表示された印刷ジョブのうち不要なものがあれば、一覧の中から選択して削除要求を出すことができる。

## 6. 評価

### 6.1 システム性能

本システムの導入によるスループット低下の要因として考えられるものは、以下の 2 点が考えられる。

- (1) 印刷ジョブに対する印刷可否の判定時間
  - (2) プリンタの総印刷枚数カウンタの読み込み時間
- まず、印刷可否の判定については、外部プログラムとして実装しているので、このプログラムの実行にかかる時間を `time(1)` コマンドを用いてユーザのべ 24,000 人で測定したところ、平均 0.3 秒/件、最大 5.00 秒、0.3 秒から 0.4 秒の間に 23,000 件 (96%) 以上含まれているという結果を得た。印刷速度が 15~20 枚/分程度のプリンタで、1 ジョブあたり平均 2~3 枚の印刷をすることを考えれば、ほぼ問題のないオーバヘッドであると考えられる。

次に総印刷枚数カウンタの読み込みについて、実装した 2 種類の方式を、表 2 の機材を用いて構築し、プリンタ本体の実際の印刷完了から、総印刷枚数を取得するまでの処理時間を評価した。

まず、“raw socket mode” の場合、利用したプリン

表 2 調査および評価に使用した機材  
Table 2 Devices for research and evaluation.

プリンタサーバ	RedHat Linux 9, LPRng
プリンタ	EPSON LP8400PS3 (raw) EPSON LP9200PS3 (raw/SNMP) FujiXerox DocuPrint C2221 (raw/SNMP) RICOH NX650S (SNMP)
ネットワーク	100baseTX

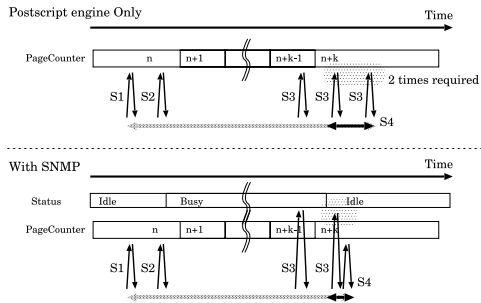


図 4 総印刷枚数取得までのタイムダイアグラム  
Fig. 4 Time diagram of read counter.

タの性能から、Step3-1 の待ち時間  $t_1$  を 10 秒、Step3-3 の待ち時間  $t_2$  を 5 秒に設定したところ、平均 7.5 秒の処理時間がかかることになる (図 4 上)。

次に、PrinterMIB を利用する場合、Step3-1 でのポーリング間隔  $t_a$  を 1 秒に設定したところ、平均 0.7 秒程度の処理時間で総印刷枚数を取得できていることが確認できた (図 4 下)。

ポーリング間隔をさらに縮めることにより、取得にかかる処理時間が短縮されることが期待できるが、使用したプリンタでは、0.1 秒以下にした場合に、SNMP による問合せが実質的に応答しない状態に陥った。これは、SNMP モジュールの CPU が過負荷状態に陥ったものと推測できる。このことから、ポーリング間隔として 1 秒を選ぶことは、SNMP モジュールに対する負荷としても適切である。

### 6.2 コスト削減効果

#### 6.2.1 システムでの総印刷枚数

図 5 は、本センターが UNIX をベースにしたシステムになった 1992 年度から 2003 年度までの、各年度におけるシステム全体の総印刷枚数と利用者数の推移をグラフにしたものである。1995 年度と 1999 年度は、システムの更新が行われたため、4 月から翌 1 月までのデータとなっている。2000 年度までは、印刷枚数制限システムがない状態であり、2001 年度以降は、本印刷枚数制限システムが稼働状態となっている。このグラフから、印刷枚数制限システムがない状態では、利用者の増加も手伝って、システム総印刷枚数が

ほぼ単調増加傾向にあった。しかし、印刷枚数制限システムが稼働を開始した 2001 年度から、若干の利用者の減少は見受けられるが、それ以上に総印刷枚数が減っていることから、本システムによって歯止めがかかっていると考えられる。

6.2.2 300 枚制限の妥当性

表 3 は、2000 年度から 2003 年度までの、前期と後期のそれぞれで利用者ごとの印刷データベースを解析した結果である。2000 年度は、利用者のモラルによる制約を期待していた状態であるが、データから、

半期に 300 枚までという制約は、それを超えている利用者数から考えて、それほど厳しいものではないことが分かる。また、300 枚を超える印刷をした利用者は 147 人と 2.5% ほどにもかかわらず、それらの利用者だけで、総印刷枚数の約 1/4 を占めていることも分かる。

2001 年度以降は、300 枚までという制限を課している状態である。本システムでは、印刷枚数上限のチェックは、印刷ジョブをプリンタに送る前にしか行えないので、印刷ジョブが大きく、結果的に制限値を超えたとしても、それは印刷終了後にしか検知できないため、若干の制限値を超えた印刷は可能になってしまう。また、300 枚を超えて印刷できなくなった場合でも、理由書を添えたうえで指導担当教員の許可を得た場合に、一度だけ 300 枚分の制限値の増加をするという運用を行っているので、300 枚を超える場合が存在している。

図 6 は、利用者ごとの印刷枚数に関するヒストグラムである。左側が前期のもの、右側が後期のものであり、50 枚ごとのランクに対して、各年度の推移を、学年別に積み上げて表現している。図 6 から分かるように、印刷枚数のヒストグラムはほぼ指数分布になっており、現在の印刷枚数制限値の 300 枚/年のところに変曲点があるようには見えない。もし、必要な印刷枚

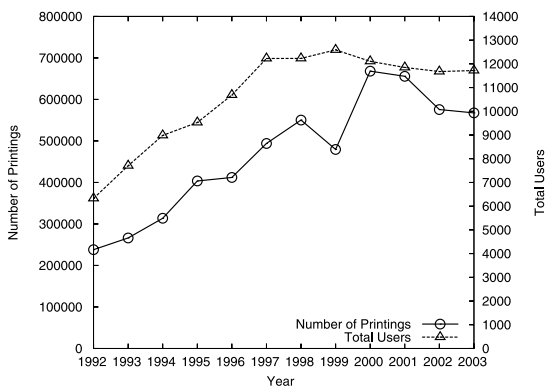


図 5 統計データ (I)

Fig. 5 Summary of the printings (I).

表 3 統計データ (II)

Table 3 Summary of the printings (II).

年度	2000		2001		2002		2003	
	前期	後期	前期	後期	前期	後期	前期	後期
印刷をした利用者の総数	5985	6130	6138	5529	5919	5025	6058	4917
総印刷枚数	328450	339606	303784	351906	283578	292136	268254	299285
利用者ごとの平均印刷枚数	54.9	55.4	49.5	63.6	47.9	58.1	44.3	60.9
制限値を超えて印刷した利用者数	147	100	31	62	45	21	20	47
	2.5%	1.6%	0.5 %	1.1%	0.8%	0.4%	0.3%	1.0%
制限値を超えて印刷された総印刷枚数	76167	43900	10854	27341	19015	7087	7080	16202
	23.2%	12.9%	3.6%	7.8%	6.7%	2.4%	2.6%	5.4%

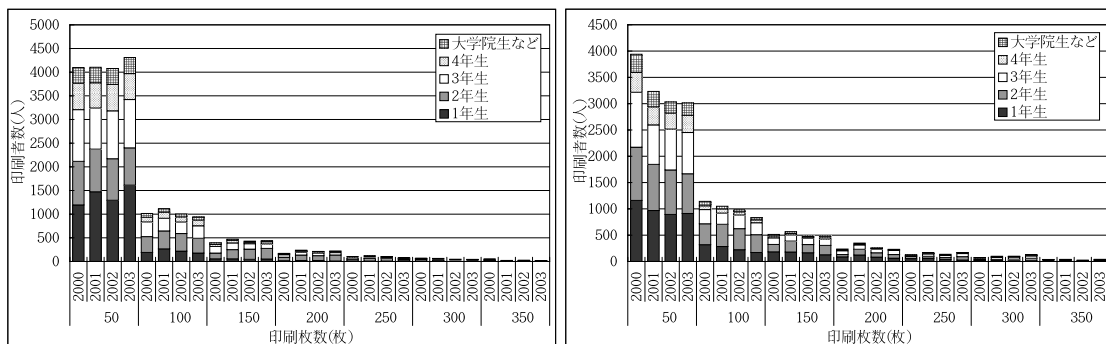


図 6 利用者ごとの印刷枚数のヒストグラム (前期 [左]・後期 [右])

Fig. 6 Histogram of the number of printing per users.



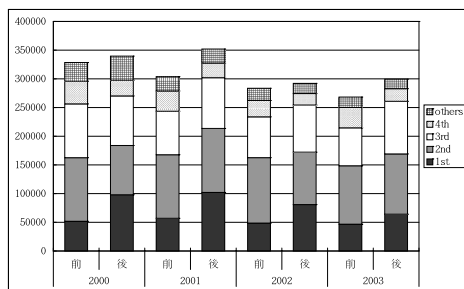


図 7 印刷枚数の学年別積み上げグラフ

Fig. 7 The number of printing per grade.

数が 300 枚に近く、無理をして抑えているとすれば、ここに変曲点がでてくると考えられる。また、典型的な年間印刷枚数の必要数値は、100~200 枚程度であると推測できる。つまり、一般の学生に無用の圧力を加えることなく（放置すると）飛びぬけて大量の印刷を行ってしまう学生にだけ制約を加えることに成功しているといえる。さらに、図 6 の 300 枚を超える印刷をした利用者の数の推移から、「300 枚の追加は一度だけ」というルールでも特に問題はないことも分かる。

### 6.2.3 運用結果の分析

2001 年度後期の印刷数の増加が見てとれるが、図 7 から、2000 年度後期の印刷数と比較すると、2 年生以上の学生の印刷数、特に 2 年生の総印刷数が増加していることが分かる。これは、2000 年度の制限枚数のない時期にシステムを利用し始めた利用者（多くは 1 年生と予想される）が、制限があることをあまり教育されていなかったために、無駄な印刷を避けようとする意識が働かなかったのではないかと推測できる。

300 枚という制限を設定したことで、300 枚を超えて印刷する人の総枚数は減少したことは確かである。2002 年度、2003 年度の後期に平均印刷枚数があまり削減されていないように見えるが、2002 年度から VMware 上で Windows 環境が使える Linux クライアントが追加されたことや、それにとまなう設置場所の統廃合、また我々の要求分析のうち、事前確認機能と取捨選択機能がこの Windows 上では統一的な形で提供できていないこと、などの要因が絡み合っていると考えられ、単純な比較対照はできない。

### 6.3 コスト負担方法に関する考察

本論文では前提として、プリンタの利用課金は利用者に直接課金しない、としたが、プリンタに対する課金としては、講義を受講するうえで一般的に十分と思われる一定枚数を基本料金（大学の場合は授業料に含まれると見なす）でまかない、それを越えた部分を受益者負担とすることが妥当であろう。本システムは国

立大学時代に開発したため受益者（学生）に現金負担を求めることが許されなかった。そのため、現金ではなく、指導担当教員からの申請書で、印刷枚数の上限を個人ベースで増やすという方式に対応して作られている。

したがって、申請書の代わりに個人の料金負担で印刷上限枚数を増やすように運用を変えれば、受益者負担での印刷体制に容易に対応することができる。ただしその場合には、窓口対応の体制作り（現金の扱いを避けるために証紙で受け付ける、卒業時点で使い残した印刷料金の返金規定あるいは返金しないとの規定など）が必要である。

## 7. ま と め

本論文では、教育用計算機システムにおける印刷システムに求められる要求をまとめ、その実装について述べた。実装したシステムは、LPRng を用いたプリントサーバと、Linux クライアント上で動作する GUI ツールからなり、プリンタからの総印刷枚数取得方式として“raw socket mode”と PrinterMIB を用いた方式の 2 種類を実装した。近年発売されるプリンタで LPR プロトコルに対応しているようなクラスのもの、知る限りすべて printerMIB に対応しており、ほとんどのプリンタに適用可能となっている。“raw socket mode”は、PrinterMIB に対応していない場合でも動作し、PrinterMIB を用いる場合は、“raw socket mode”と比較して処理時間が大幅に削減できた。

また、要求をもとに、我々のシステム上で実装したものを運用した結果、年間の総印刷枚数が 2000 年度の 67 万枚から、65 万枚、58 万枚、57 万枚と減少しており、総印刷枚数が削減されたことを確認した。また、印刷枚数を節約できるためのツール群を用意したうえで 300 枚の印刷枚数制限を行うことで、一般の学生に無用の圧力を加えることなく、飛びぬけて大量の印刷を行ってしまう学生にだけ制約を加えることに成功した。

本論文では、実装に際して、Linux クライアントを用い、Postscript を用いているが、クライアント計算機として Windows を用いた場合でも、LPD を利用してプリントジョブをプリンタサーバに送付すれば、PrinterMIB を用いて印刷枚数をカウントし制限する部分は、そのまま適用可能であると考えられる。

今後の課題として、さらに汎用性を高めるために、プリンタ機種ごとの PrinterMIB の表現値の差異を吸収する設定ツールの拡充や、単なる総印刷枚数制限だけでなく、両面印刷、モノクロとカラーの違いなどの

プリンタ種別による差異や、授業時間内の印刷と自習時間での印刷を別のくりにして算定するといった、より詳細なアカウント管理を可能とする統合管理システムを構築していくこと、また Postscript 以外を使用する環境 (Windows) での実現方法の検討があげられる。

### 参 考 文 献

- 1) 田中哲朗, 安東孝二, 吉岡 顕: 複数 OS 環境におけるユーザ管理, 情処技研 (DSM), 16-9, pp.49-54 (1999.11).
- 2) 丸山 伸, 藤井康雄, 中村順一: 出力制限の可能な印刷システムの構築と運用平成 13 年度情報処理教育研究集会, p.342 (2001.10).
- 3) 小川剛史, 中村匡秀, 近藤弘一, 大崎博之, 榎田秀夫, 北道淳司, 中西通雄: Linux システムにおける授業・運用支援系ツールの開発, 平成 12 年度情報処理教育研究集会, pp.239-242 (2000.12).
- 4) Masuda, H., Nakamura, M., Kondo, K., Saitoh, A. and Nakanishi, M.: Requirements and implementation of printing subsystem in educational computer system, *Pan-Yellow-Sea International Workshop on Information Technologies for Network Era (PYIWIT2002)*, pp.251-256 (Mar. 2002).
- 5) 榎田秀夫, 齊藤明紀, 中宮広揮, 増澤利光: SNMP を用いた汎用プリント枚数取得システム, 情報科学技術レターズ (FIT2004), LL-003, pp.317-318 (2004.09).
- 6) Smith, R., Wright, F., Hastings, T., Zilles, S. and Gyllenskog, J.: PrinterMIB v2, RFC1759 (1995).
- 7) Bergman, R., Lewis, H. and McDonald, I.: PrinterMIB v2, RFC3805 (2004).
- 8) リコー: Ricoh IO Gate.  
[http://www.rioh.co.jp/IPSiO/related\\_goods/iogate/](http://www.rioh.co.jp/IPSiO/related_goods/iogate/)
- 9) 富士ゼロックス: DocuHouse.  
<http://www.fujixerox.co.jp/soft/docuhouse/>

(平成 16 年 7 月 12 日受付)

(平成 17 年 2 月 1 日採録)



榎田 秀夫 (正会員)

1998 年大阪大学大学院基礎工学研究科博士後期課程修了。同年大阪大学情報処理教育センター助手, 2000 年同大学サイバーメディアセンター情報メディア教育研究部門助手となり,

現在に至る。博士 (工学)。電子情報通信学会会員。



小川 剛史 (正会員)

1999 年大阪大学大学院工学研究科博士前期課程修了。2000 年同工学研究科博士後期課程中退後, 大阪大学サイバーメディアセンター情報メディア教育研究部門助手となり,

現在に至る。博士 (情報科学)。ACM, 電子情報通信学会, 日本バーチャルリアリティ学会各会員。



齊藤 明紀 (正会員)

1991 年大阪大学大学院博士後期課程修了。大阪大学基礎工学部助手, 同大学情報処理教育センター講師, 同基礎工学研究科講師, 基礎工学研究科助教授を経て, 2004 年より鳥取

環境大学情報システム学科教授。工学博士。電子情報通信学会会員。



中村 匡秀 (正会員)

1999 年大阪大学大学院博士後期課程修了。同年カナダ・オタワ大学ポスドクフェロー。2000 年大阪大学サイバーメディアセンター助手。2002 年奈良先端科学技術大学院大学情報

科学研究科助手となり, 現在に至る。博士 (工学)。電子情報通信学会, IEEE 各会員。



近藤 弘一

1997 年同志社大学大学院工学研究科電気工学専攻修了。1999 年大阪大学大学院基礎工学研究科数理科学分野博士後期課程を中退後, 同年大阪大学基礎工学研究科助手および

同大学サイバーメディアセンター助手を兼任。2002 年同志社大学工学部電気工学科専任講師となり, 現在に至る。博士 (工学)。日本数学会, 日本応用数理学会, 日本物理学会, SIAM 各会員。



中西 通雄 (正会員)

1980 年大阪大学大学院博士前期課程修了。三菱電機, 大阪大学基礎工学部助手, 同大学情報処理教育センター/サイバーメディアセンター助教授を経て, 2002 年より大阪工

業大学情報科学部教授。放送大学客員教授。