

HITAC 8350 EDOS 出力スプール・プログラム

鎌田 顕一* , 磯田 和男*

(昭和50年9月8日受理)

HITAC 8350 EDOS Output SPOOL Program

KEN-ICHI KAMATA and KAZUO ISODA

Abstract:—New WRT6 is an output SPOOL program which has been developed by the Information Science Department at Ibaraki University. This program is controlled by the Executive program of HITAC 8350 EDOS.

In order to obtain maximum performance the program uses a buffering method and overlaps disk retrieval and printer output.

It has effected a reduction of up to about 16 percent in the running time for printout in comparison with the conventional output SPOOL program OUTWRT. In addition, use of the program has decreased operator's intervention caused by the printer troubles. It requires only 4K bytes; the more memory will be available for user problem programs.

1. ま え が き

マルチプログラミングの技術の発達とともに、スプール (SPOOL; Simultaneous Peripheral Operation On-Line) システムが考えられ、電子計算機による処理が効率的に行なわれるようになった。これはユーザーに最も頻繁に使用されるカード・リーダーやプリンターなどの入出力装置の速度が遅いために、その処理効率が低下することを防ぐ目的で考えられたシステムで、それらの代りに仮想的に速度の早い磁気テープ、ディスクあるいはドラムなどの装置に対して入出力を行なうものである。そのためには、ジョブの実行に先立って、カード・デッキをそれらの仮想入出力装置に読みこんでおく必要があり、また出力はジョブが終了しなければ得られないという欠点がある。

しかしながらジョブの処理とこのスプール・プログラムを並行して実行させて、連続的にジョブを実行するならば、それらのターン・アラウンド・タイムをかなり小

さくさせることができる。

さらにもう一つの利点がある。ディスクやドラムは領域を分割して使うことができるので、いくつかのジョブに対して同時にサービスを行なうことができる。ところが、カード・リーダーやプリンターなどの装置は一つのジョブに占有されてしまうので、もしそれを必要とする他のジョブがマルチプログラミング・システムで実行されるとすれば、そのジョブの数に等しい台数の装置が必要となるであろう。これは装置の通常の使い方からみて不経済なことである。この問題はやはりスプール・システムにより解決され、それぞれのジョブは必要とする入出力装置を仮想的にこれらの装置の領域に割り当てて占有することができるようになっていく。この技術は最初に Atlas システム¹⁾にとり入れられ、その後いろいろなシステムに数多く採用されている。

本学情報工学科に設置されている HITAC 8350 の EDOS (Extended Disk Operating System) にもこの目的のために、入力リーダーおよび出力ライター

*茨城大学工学部情報工学科 (日上市中成沢町)

と呼ばれるスプール・システムが用意されている。²⁾この出力ライター・プログラム OUTWRT にはいろいろな優れた機能が備わっており、ユーザーの便宜をはかっているけれども、学生の教育・演習を主とするように運営されているセンターではいくつかの不都合な点があるようである。

その一つにプリンターの印刷速度の問題がある。授業時間という限られた時間内に数多くの処理を行なわなければならないところでは、便利であっても、めったに使用されない機能のために印刷速度が低下することは望ましくない。

もう一つの問題は、特に専任のオペレータなしに運営されるセンターでは、できるだけ簡単なオペレーションが要求されるということである。例えば、フォームのジャム、あるいはオペレーション・ミスなどによって生じるエラーを回復するのに物理的な回復に加えて、コンソールからメッセージを応答しなければならないということは、複雑なオペレーションということができよう。そのためにオペレーションのマニュアルを調べて、正しい応答のフォーマットを探し出すことは、学生のような不慣れたオペレータにとって、相当なロードとなるからである。

それらはターン・アラウンド・タイムを劣化させる結果となる。これらの二つの問題を解決するために、ファコム・ハイタック(株)の SE の方々のご協力を得て、新しい出力ライター・プログラム WRT6 を開発することができた。ここでは、この WRT6 の機能と構成について概略を述べる。

WRT6 は EDOS の制御の下に実行されるようになっているため、この名を持っている。EDOS の出力ライター・プログラムは OUTWRT かまたは WRT6 というプログラム名を持たねばならないからである。

2 機能と構成

情報工学科に設置されている、HITAC8350 電子計算システムの機械構成は Fig. 1 に示すとおりで、EDOS と呼ばれるオペレーティング・システムによって制御されている。EDOS で実行されるジョブは SYSOUT と呼ばれるディスク・ファイルにすべて登録される。^{*}つまりプリンターに対するすべての出力は、仮想的にこの SYSOUT ファイルにスタックされるようになっている。ジ

ョブが終了すると、この出力スプール・プログラム WRT6 はスタックされたデータを最大限のスピードで、プリンターに出力するように設計されている。

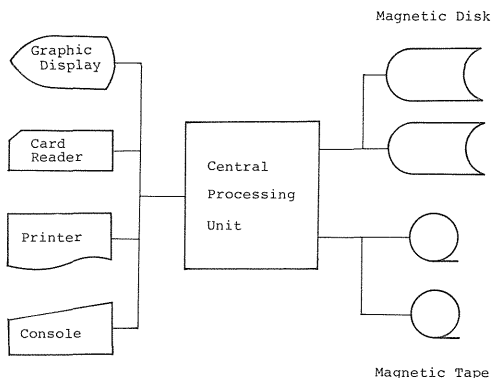


Fig. 1 System Configuration.

WRT6 は EDOS エグゼクティブ・プログラム監視の下で、他の処理プログラム、ユーティリティ・プログラムあるいは入力リーダー・プログラムなどと並行して実行される。これらの関係を Fig. 2 に示す。

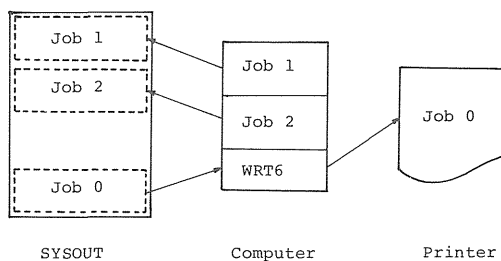


Fig. 2 Output SPOOL System.

この図では、二つのジョブと WRT6 がマルチプログラミング・システムで実行されている。SYSOUT ファイルは各ジョブに対して分割されていて、両方のジョブ 1、および 2 とともに同時に出力を行なうことができる。WRT6 は既に処理の行なわれたジョブ 0 の出力を印刷している。

WRT6 が出力するとき、最大限の印刷速度を得るためには、ディスクの読みとり、データの印刷のような入出力オペレーションを効率良く行なうことが必要である。その方法として、ダブル・バッファリング³⁾の考え方があ

*この方法はオプションであり、SYSOUT ファイルによらないで直接プリンターに出力する方法もある。

ロックされていないレコードの入出力に適しているので、この場合にはそのまま利用できない。SYSOUTファイルのデータは可変長のブロックされたレコードであるから。そこでそれを応用した方法を用いてある。Fig. 3にそれらの概略的な流れ図を示す。この図では最後のレコードの処理は省略されている。

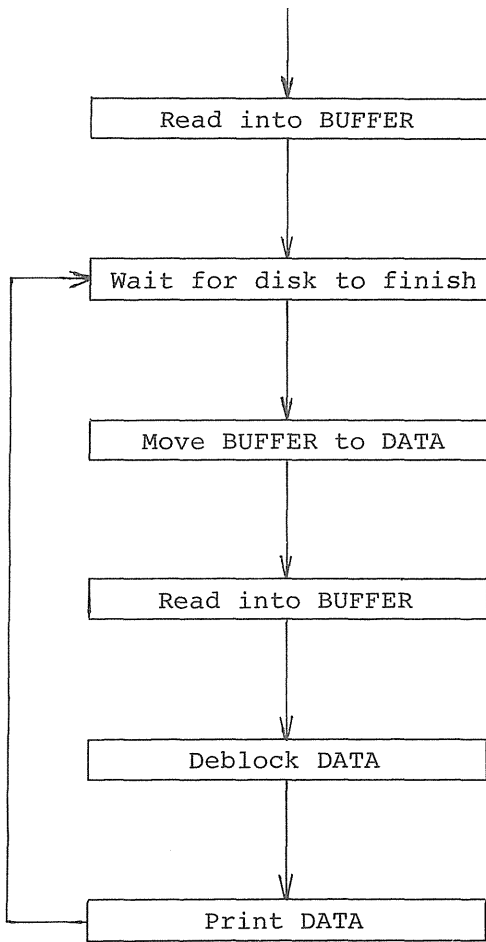


Fig. 3 L/O operation with buffering.

最初に一つのフィジカル・レコードを読みとる。読みとりが完了するのを待って次のフィジカル・レコードの読みとり命令をセクター・チャンネルに与える。この命令の完了を待たないで、既に読みこまれているレコードのデブロッキングを行ない、プリンターに出力する。これは最後のロジカル・レコードまで繰り返して行なわれる。それは既に読みとりの命令が与えられているので、そのオペレーションの完了を待つ命令にコントロールが

移される。

フォームのジャム、オペレーション・ミスあるいはエンド・オブ・フォームなどによって、プリンターにトラブルが生じたときには、メッセージがコンソール・ディスプレイに表示され、回復されるまで、プログラムはアイドルとなる。オペレータはトラブルの原因を取り除くだけでなく、プログラムは次の処理を続行する。

WRT6のプログラムは九つの処理ルーティンと定数および変数のための領域とから構成されている。全体のコア容量は、4 Kバイト以内に収まるように設計されている。プログラムは単一のモジュールから成り、オーバーレイを行なっていないが、ファイルのオープン・ルーティンは最初にロードされたときのみ使われるだけで、その後は不要となるために、入出力用のデータ領域として使われ、コア容量の節約をはかっている。

各ルーティンの機能は次のとおりで、Fig. 4にこれらの概略を示す。この図では、オペレータ・リクエストの処理ルーティンは省略してある。

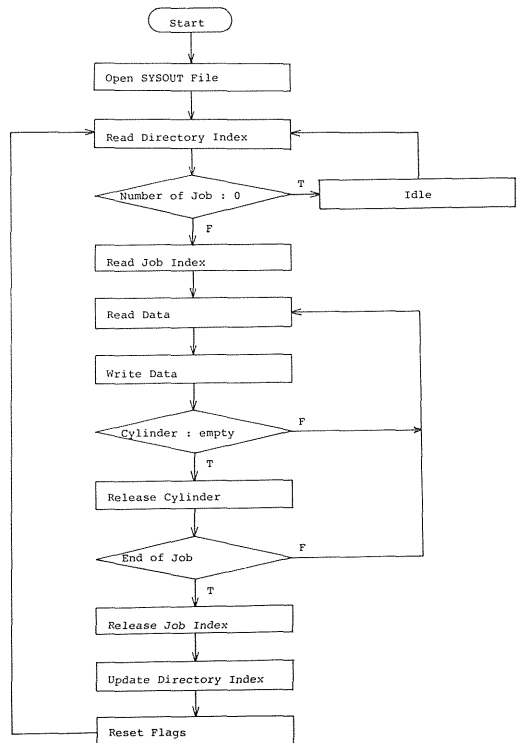


Fig. 4 General Flowchart

(1) SYSOUTファイルのオープン

SYSOUTファイルのディスク・アドレスを得るために、まず最初に標準ボリューム・ラベルを読みとり、VTOC (Volume Table of Contents)のアドレスを得る必要がある。これはファイル・ディレクトリとも呼ばれ、それにはそのボリュームに含まれるファイルの名称、長さおよびアドレスなどが書かれている。次にそのアドレスによって、VTOCを読みとり、SYSOUTファイルのディスク・アドレスを得ることができる。

(2) ディレクトリ・インデックスの読みとり

SYSOUTファイルのディレクトリ・インデックスはジョブ・インデックス(ディレクトリ)として指定されている領域のアドレスや現在スタックされているジョブの数を含み、その長さは80バイトである。ここではそのジョブ数を調べ零でないときには、それらを順番にプリンターに出力することになる。もし零であるときには、コントロールを他のパーティションのプログラムに移し、WRT6は準備のできるまで、アイドル状態となる。

(3) ジョブ・インデックスの読みとり

ジョブ数が零でない場合には、ディレクトリ・インデックスにあるディスク・アドレスにより、ジョブ・インデックスを読みとる。ジョブ・インデックスは、スタックされているジョブの数だけあり、その長さはまた80バイトである。これはジョブの出力の状態を表わすジョブ・フラッグやジョブ番号、ジョブ名、ジョブ・クラスなどのジョブに関する情報、および出力データのディスク・アドレスが含まれている。

(4) 出力データの読みとりおよび印刷

ジョブ・インデックスに含まれるディスク・アドレスにより、出力データが読みとられ、実際にプリンターに出力される。このデータは可変長のブロックされたレコードであり、デブロッキングを行なわねばならない。

(5) シリンダーの開放

EDOSは出力データをディスクにスタックするが、このための領域は循環的に使われる。そのために使用中の領域と未使用中のそれとの区別をしなければならない。このときの単位はシリンダーである。

スタックされたデータを出力し終えたシリンダーは開放される。これはシリンダー・マトリックスの対応するビットをリセットすることによって行なわれる。

一つのジョブがすべて出力されると、次の三つのルーティンが実行される。

(6) ジョブ・インデックスの開放

ジョブ・インデックスを開放するために、ジョブ・フラッグをリセットする。

(7) ディレクトリ・インデックスの更新

ディレクトリ・インデックスにあるジョブ・インデックスの数を更新する。

(8) フラッグのリセット

プログラム実行中に必要なエラー・フラッグとオペレータ・リクエストのために使われるフラッグなどをリセットする。

オペレータが指示できる機能は次の三つである。

(a) データの印刷中止。

(b) 印刷のやり直し。

(c) プログラムの終了。

これらは次の処理ルーティンで行なわれる。

(9) オペレータ・リクエストの処理

不要なデータの印刷中止、フォームのジャムなどによる印刷のやり直しおよびWRT6の終了はオペレータの指示によって行なわれる。それはコンソール・キーボードからメッセージをタイプすることによってなされるが、そのフォーマットはOUTWRTの場合と同じようになっている。それはオペレータの混乱を少くするためである。

次に印刷速度について、OUTWRTとの比較を試みる。Table 1は英数字および特殊文字を1000行印刷したときに要する時間を示している。これらは文字数に関係なく一定であり、WRT6はOUTWRTの約84パーセントとなっている。Table 2は同じように12文字を印刷したときの時間を示している。Table 1および2から、ソース・プログラムのような特殊文字を含む印刷のときには、WRT6は最も時間を短縮することが分かる。また計算結果のようなデータ印刷では、わずかではあるが、やはりWRT6の方が早い。実際のジョブについて調べてみよう。Table 3は、学生のFORTRANとアセンブラ・プログラムのジョブを印刷するのに必要な時間を示している。これらの結果、約6パーセント以上の時間を節約できることが分かる。

Table 1 Run time for printout of the alphameric and the special characters.

Number of characters per line	Run time [sec]		Ratio [%]
	WRT6	OUTWRT	
132	84.5	101.2	83.5
112	84.4	101.3	83.3
92	84.5	101.1	83.6
72	84.7	101.6	83.4
52	84.6	101.4	83.4
32	84.9	101.7	83.5
12	84.8	101.3	83.7

Table 2 Run time for printout of twelve characters.

Kind of characters	Run time [sec]		Ratio [%]
	WRT6	OUTWRT	
Alphameric	77.3	85.3	90.6
Alphabetic	62.0	62.9	98.6
Numeric	60.1	60.2	99.8

Table 3 Run time for printout of job.

Program	Run time [sec]		Ratio [%]
	WRT6	OUTWRT	
FORTRAN	63.5	67.6	94.0
Assembler	45.7	49.5	92.3

3. あとがき

出力ライター・プログラム WRT6 について、その概要を述べた。これはメーカーの OUTWRT に比べて、次のような点が優れているといえる。

- (1) 印刷速度の早いこと。
- (2) トラブルのエラー回復手順の簡単なこと。

- (3) コア容量が小さく、4 K 以内であること。

印刷速度は OUTWRT に比べたところ最高約 16 パーセントの時間が節約できるようになった。これは入出力オペレーションのオーバーラッピングを行ない、さらにスピード・アップをはかるために、次のような機能を省略したためである。これらは便利な機能であるが、通常はほとんど使用されないからである。

- (1) 出力タイプの指定によるフォーム交換。
- (2) 出力データの試し印刷。
- (3) データの複数回印刷。
- (4) ジョブ・コントロール・ステートメントによる印刷制御

最後にあたり、この WRT6 の開発にあたっていろいろ援助して下されたファコム・ハイタック㈱の里川および綾部両主任にお礼を申し上げる。さらに同社の元 S E の故田辺氏には細いアドバイスを頂いた。ご逝去の訃音に接し、ここに謹んで哀悼の意を表したい。また新潟大学情報処理センターの石垣氏には、同氏の開発された WRT6 を参考にさせて頂き、ここに厚く感謝を申し上げる。またプログラム・カード穿孔には、本学情報工学科の本多令子氏、またプログラム・テストには、同学科の谷川邦夫氏にお世話になった。合わせてお礼を申し上げたい。

参 考 文 献

- 1) T. Kilburn, D. J. Howarth, R. B. Payne, F. H. Sumner: Computer Journal 4, 3, pp. 222 - 225, Oct (1961).
- 2) ㈱日立製作所: HITAC 8350 EDOS コントロール・システム, 8300-3-100, pp 119-154, (1975)
- 3) S. E. Madnick, J. J. Donovan: Operating Systems, McGraw-Hill, Inc., p. 64, (1974)