マルチアクセス計算機システム*

Maurice V. Wilkes**
訳：高　橋　　茂***

会場の皆さま．この講演を依頼されたことは私にとってこの上もない名誉であるということを最初にいわせて頂きたい．と申しますのは日本は計算機が広く使われている国であり，また計算機工業が繁栄しつつある数少ない国の一つだからであります．さらに，日本は計算機技術に創意に富む貢献をしてきた国であります．

初期の計算機はユーザ自身によって操作されたものでありまして，“hands-on”とでも呼びますか，そういう使い方をしておりましたが，これはユーザにとって極めて満足なものでありました．これらのユーザは計算機を無制限に，1時間でも，使うことができ，その間に存分に仕事を進めることができたものです．

しかしながら，計算機というものは一人の人間に独占させるには，当時としてはあまりにも数が少なくかつ貴重なものであることが間もなく明らかになり，人間を働かせる効率よりも，機械を働かせる効率を重視する時期が始まったのであります．

そこで，操作手順の効率を強調したシステム，すなわちバッチ処理システムが開発されたのであります．このシステムでは，仕事は一まとめにして計算機に与えられ，処理されるわけで，機械の使用効率は向上されましたが，ユーザは全くうんざりさせられる結果となりました．

最近，小形の安価な計算機が得られるようになり，世代の違う計算機ユーザが“hands-on”式の使い方を楽しんでいるのは喜ばしいことであります．しかしながら，これは完全な解答ではありません．というのは小形計算機はその能力が極めて限られており，計算機のユーザ，特に計算機を気軽に使おうというユーザが現在期待しているような強力なソフトウェアによる助力を提供するものではないからであります．そこで，通常マルチアクセスあるいはタイムシェアリングと呼ばれているもののなかに私たちが見出しつつあるものが，正しい解答であると私は思います．

これらのシステムではユーザは，多分自分の部屋に設置してあって，電線で計算機に接続してあるコンソール（通常テレタイプ）のそばに坐っており，計算機はその時間を多くのユーザに分配するのであります．ユーザの一人が考えようとしていて，あるいは他の理由で一寸休むと，計算機は他の人のために仕事をします．事実，計算機は，一つの仕事から他の仕事にすばやく移動します．計算機がある仕事を中断されると，それをどこまでやったかを覚えていて，順当に行けば，またそこへ戻ってきます．各ユーザは機械をあたかも占有しているかのように感じます．もちろん，彼は機械を他のユーザと共用しているわけですから，本当に占有しているよりは，それが遅く見えることになるのですが．

いまのところタイムシェアリングは計算機の世界を襲った一連の革命のなかでの最新のものであります．この20年間，計算機の進歩は極めて急速なものでありましたが，この進歩は一定の歩調のものではなく，むしろ革命につぐ革命という調子のものでありました．私たちの憶えている革命をいくつか挙げてみますと，Fortran, Algol などの上位レベルのプログラム言語の開発，計算機技術者から見て一大革命であったところのトランジスタの出現，また計算機への割込みを可能にした技術的な方式の開発，さらに大容量磁気ディスクの開発などがあります．

ところで，タイムシェアリングを可能にしたのは新らしい技術的な発明ではなく，むしろすでに存在していた技術的な手段を，一大前進のために現実化したということではないかと私は考えます．

タイムシェアリングシステムの開発という仕事はその大部分がソフトウェア作りでありますが，同時に若干のハードウェアの開発を必要としております．現存する計算機でタイムシェアリングに全面的に適したも

* Multi-Access Computer System, by M.V. Wilkes (F.R.S., University Mathematical Laboratory, Cambridge, England). 昭和43年4月2日，日経ホールにおける日本情報処理開発センター，日本電子工業振興協会と共催の講演会での講演．

** 英国ケンブリッジ大学教授，数学研究所長，英王室協会会員
*** (株)日立製作所神奈川工場開発部

のはないといってよいのではないかと私は思います．もちろん次期計算機として生産されるものはこれに適するものと確信しておりますが．

皆さま方のなかにはタイムシェアリングシステムを使った経験のない方もおられると思いますので，1年前，私たちのケンブリッジのシステムが開発中であったころに撮った短いフィルムをお目にかけたいと思います．このフィルムについて多少弁解しておかねばならないことは，これが手製のものだということであります．この機会に間に合うようにましなものを作らせたいと思い，多少努力致しましたが，とうとう出発前には間に合いませんでした．

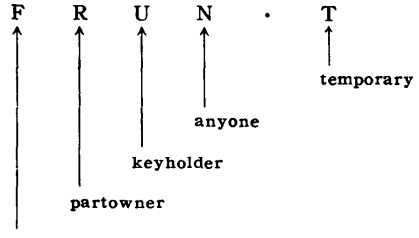
大形のタイムシェアリングシステムに欠くことのできない一つの特色は，磁気ディスクに情報を保存しておけるというファイリングシステムについての計算機の手段であります．そこで，最初のスライドをお願いして，あとでフィルムでお目にかける事柄のうち，ファイリングシステムの働きについて特に説明しておきたいと思います．

さて，このシステムではユーザがある情報をタイプし，これをファイルとして自分で選んだ名前をつけて納うことができます．この名前は三つの部分からなり，第1の部分は実はユーザの選択ではなく，このシステムのユーザとしての彼自身の認識記号（identification）であり，これに続く二つの部分が自由に選べるのであります．大きなシステムでは，許可なしに他人が自分のファイルに手を触れないように，ユーザがファイルを保護できるようにしておく必要があります．そこで，ユーザは，もしそうしたければ，名前のあとに自分以外の誰かが，いかなる条件の下で，自分のファイルに手を触れる（アクセスする）ことができるかという情報をもつセクションをタイプすることができます．最初の文字はユーザすなわちファイルの所有者が自分自身に与える特権を示すもので，この例では“自由に（freely）”アクセスできることを示しています．第2の文字はユーザが自分のパートオーナ（partowner），すなわちパートオーナリストに彼が挙げている何人かの人の一人に，彼が与える特権を示しています．この場合パートオーナは“読むだけ（read only）”であります．キーホールダ（keyholder）というのはそれをタイプすればファイルにアクセスできる秘密のキーワード（keyword）をユーザから委されている人間で，この例ではキーホールダはファイルを更新してよいことになっています．すなわち，彼はファイルに何か付け加えても，またそれを変更してもよいことになっています．最後の文字は不特定の他人に許される権利を示すもので，この場合にはノーアクセス（no access）となっています．

さらにそのあとに，いま1字，ファイルが一時的（temporary）なものであるか，スライドの例はそうな

Filing System

MVW731/INTEREST/VERSION1




```
EXAMINE
WHICH MODE? FULL

FILES FOR MVW731/        AT 15.47.25. ON 2 1  67
/TEST/A/FFNN. T                1     17.44 DEC 12   17.38 DEC 12
/TEST/B/FNNN. P                1 D   16.46 OCT 21    9.44 OCT 12
END

READY
```

Slide No. 1

のですが，あるいは永久的（permanent）であるかを示す字があります．ここにPとあれば，ファイルが永久的であることを示し，システムはこの場合，ファイルが障害を受けた場合に備えてこれを磁気テープにコピーし，ファイルが再生されるようにしておきます．すなわち，この場合には，情報を安全に保つ責任はユーザではなくシステムにあることになります．

もしユーザが自分が持っているファイルのリストであるところのファイルのディレクトリー（directory）を調べたいと思えば，彼は EXAMIN というコマンドをタイプします．システムは WHICH MODE？とタイプすることによって応答し，この場合ユーザは FULL とタイプして，彼がファイルディレクトリーを全部打出したいということを示しています．そしてこのスライドで御覧のように，ここにはただ二つのファイル，一時的なものと永久的なものがあるだけです．ここでDという文字は安全のために磁気テープにコピーをとっておく操作がすでに実施されたことを示しています．

もしユーザが望むならば，題名だけ，という意味の TITLES という語をタイプすることができます．もし彼が CHECK FULL あるいは CHECK TITLES とタイプすれば，計算機は各行の終りでタイプを中断し，ユーザはそこで命令をタイプすることができます．たとえば彼はファイルを削除したり，そのステータス(status)を変更したり，あるいは印字のモードを変更したりすることができます．申し忘れましたが，もしユーザがさきに説明したステータスを表わす文字をタイプしなければ，情報不足の処置がとられる，すなわち，計算機はユーザがある定まった文字の組合せを要求しているものと見做すのです．またユーザが自分のファイルを使っているときには，彼は第1の部分すなわち彼の認識記号をタイプする必要がないということも申上げておくべきでした．

次のスライドをお願いします．

ユーザがファイルの変更，あるいはいまタイプして入れたばかりの情報の変更を行ないたい場合，彼は EDIT という語をタイプします．これはエディタ（editor）を呼び出し，ユーザはこれにいろいろなことを頼めるわけです．スライドにこのようなユーザの要求をいくつか示しておきました．Fは“find”を意味し，ALPHA で始まる最初の行を見付けます．ポインタというものがあって，ファイルの最初の行から下の行へと下りて行くというように想像すればよいでしょう．F. ALPHA はこのポインタを ALPHA で始まる行まで下ろして，ユーザが調べられるようにその行をタイプで打ち出します．L. ALPHA も同様ですが，この場合は ALPHA は行の始めになくてもよいのです．

The EDIT command

F. ALPHA.	Find next line beginning with ALPHA
L. ALPHA.	Locate next line containing ALPHA
A. ALPHA. BETA.	Insert BETA after ALPHA in current line
N. n.	Move pointer down n lines
M. n.	Ditto but delete lines
Q	Move pointer to end of file and quit

Slide No. 2

ユーザが変更したいと思う行をポインタが指示しているときに，ユーザはスライドに示すようにいろいろな要求をエディタにすることができます．Aは“after”の意味で，ここでは ALPHA のあとに BETA を挿入することを意味します．いまお目にかけるフィルムではこれに似た他の要求が例示されています．さてユーザが編集を終わると，彼は QUIT を意味するQという字をタイプし，これによってポインタはファイルの終りまで動いて行き，編集の操作は終りになります．フィルムではユーザが一つのプログラムをタイプして入れるところをお見せしますが，これは私たちがケンブリッジで使っている AUTOCODE と呼ばれる

COMPOUND INTEREST

(Program written in AUTOCODE, a local language)

```
PROGRAM  0
1: LAYOUT  0 : 20 : 4
CRLF
TEXT: RATE OF INTEREST=:
X=INPUT
A=100,
FOR  N=1 : 1 : 10
  A=A (1+X/100)
  PRINT (N) 2 : 0
  PRINT (A) 2 : 4
REPEAT
→ 1
START 1
```

Slide No. 3

一つの方言で書かれたプログラムでありまして，彼はそれを打ち終ったとき，それをファイルしますが，それは計算機には現用のドキュメント（current document）として残されています．そこで AUTOCODE というコマンドをタイプすることによって，ユーザはこれをコンパイルさせ実行させることができます．プログラムが実行されますと，ユーザは結果をもらうか，あるいはもし間違いを犯していたならば，診断の結果をもらうかのどちらかです．では次のスライドをお願いします．

これがタイプで打ち込まれたプログラムです．これについていま詳細にやって見るつもりはありません．これは複利表の計算，すなわち 10 年間に亘って各年末での金額の計算を行なうためのプログラムです．“X＝INPUT”という命令がありますが，ここで機械はユーザが利率をタイプするのを待ち，それがタイプされると引続きプログラムを実行し，100 という金額で始まる計算を行ないます．ここに表を打ち出すループがあります．

さてフィルムについて一，二つけ加えておきますと，このフィルムを作りましたときには，面倒なログイン（log-in）の手順というものはありませんでした．したがってユーザは単にコンソールのところへ行って，復帰のキーを押すだけです．いま一つ申上げておいた方がよいと思いますのは，オペレータがテレプリンタのカバーをはずすのを御覧になるでしょうが，これはカメラに都合よくしているので他意はないということです．

さてフィルムをお願いします．これは無声のフィルムで多分 1920 年代の始めの頃に流行していたような説明付きのものでございます．

——映画（約5分間）——

さて，この映画によって私たちのシステムがどういったようなものかということについて多少お判り頂けたことと思います．このシステムは昨年中大いに開発され，いまでは多くの特長がつけ加えられました．

タイムシェアリングシステムの設計にとり入れるべき，いくつかの考慮について簡単に説明させて頂きたいと思います．システムはコンソールのそばに坐っている人々の仕事，これをフォーグラウンド（foreground）と呼びますが，このフォーグラウンドの仕事の実行を要求されるだけではなく，通常の方法で計算機にオフラインで与えられるバックグラウンド（background）の仕事を実行することをも期待されています．

初期のシステムではフォーグラウンドとバックグラウンドとは全く分離されていましたが，今日ではフォーグラウンドとバックグラウンドを一体化して，それによってユーザがコンソールから自分のファイルにアクセスして，仕事をオンラインでもオフラインでも都合のよい方で実行させることができるようにするという試みがなされています．あるユーザは他のユーザよりもプログラムとより密接により頻繁にやりとりを必要とします．実際ある人々は完全に会話的な操作モードを必要としていますが，このモードではプログラムはユーザに情報をタイプし，次にユーザがプログラムに対して情報をタイプするという具合にして，プログラムが実行されて行きます．

他方，ある人々はプログラムが実行されている間に，それから情報を受けとることはしたいが，それに情報をタイプして入れることは必要としないのです．このような人々は，私たちのシステムでは，RUNJOB というコマンドをタイプします．そうしますと，計算機は仕事を実行し，結果をラインプリンタに出すか，あるいは結果をユーザが次にログインしたときにアクセスできるファイルに入れておきます．またユーザによってはコンソールを全く使わないでオフラインで仕事を実行させることで全く満足だという場合もあります．

これらの異なる要求があるということを積極的に利用すること，また個々のユーザに欲するところのものは与えるが，欲する以上のものを与えてリソース（resources）を浪費しないようにすることは，システム設計者の狙いであります．実際タイムシェアリングを可能にしているのは，ある時点で見るときユーザ間に要求の相違があるということでありまして，もしコンソールからログインしている人が，だれもかも沢山の計算機時間を要するようなプログラムを走らせようとしたと致しますと，システムは面倒を見切れないことになってしまうでしょう．

さてこれまでは，システムの開発ということに注意が集中されていたのですが，いまや沢山のシステムが稼動しておりますので，タイムシェアリングシステムの管理という問題にかなりの注意が向けられるようになりました．もちろん計算センターの管理の問題は永年私たちが付き合ってきた問題ですが，それでも完全

に解決された問題とは申せません．タイムシェアリングはこれにさらに問題を付け加えるものであります．

タイムシェアリングシステムではユーザは実質的にはシステムの外ではなくて，内にいるので，システムがあたかも実時間であるかのように，ユーザを制御する必要があります．計算機のマネージャのこのような仕事を可能にするために欠かせてはならないことは，システム設計者がそのための道具を提供すべきだということです．ケンブリッジにおいて私たちは，システムが提供するすべてのリソースを綿密に制御するための手段をマネージャに与えるべきだという見解をとってきました．リソースとは，計算機時間，コンソール時間，メモリースペース，ファイルスペース，それに現在使っていないファイルを保管しておくための磁気テープ上のスペースです．ユーザはこれら種々のリソースの割当てを受けるわけですが，割当ては全体でどれだけという場合と，各シフトに対して別々にどれだけという場合とがあります．

主要なプログラムの一つ，というよりはシステムの運転を制御するスーパバイザの主要なルーチンの一つは，ログインルーチン（logging-in routine）であります．このルーチンはユーザが最初にログインしたときにユーザの認識記号をチェックする責任をももっていて，ユーザに名前とプロジェクト番号をたずね，さらにユーザにパスワードをタイプするように頼みます．私たちのシステムでは，ユーザは自分自身のパスワードを選び，それをいつでも変えられるようになっています．タイムシェアリングについての論議で，しばしば非常な興味をもたれるものに，システムの保護，すなわち許可をもたない人たちがアクセスすることを防ぐということがあります．この保護ということは組織体によってその重要性に差異があるでしょう．しかし大学のようなところでさえ，学生が他人の名前のもとにログインしたり，他の人々のファイルに干渉したりし得ないようにしておくことが望まれます．また学生たちがシステムの記録にアクセスして自分たちの計算機時間の割当てを増やしたりできないようにしておくということは非常に大切なことであります．

システムに対して許可されないものがアクセスできないようにするとか，コンソールからの操作の妨害にならないようにするということは，システムの設計者が自身に課するにもっともな目標だと私は思います．私たちのシステムでは，現在どうなっているかというと，ユーザはログインするときに，彼がシステムのリソースとして必要なもの，すなわちどれだけメモリーを使う必要があるか，どれだけ計算機時間を使うか，などということを記述しなければならないのです．そしてもしシステムが彼に要求されただけのものを与えられないならば，与えられるだけのものを与えるというようになっています．ユーザが優先権をもっていることがあり，ログインに際してユーザはそれを使うことができます．しかしながら，高い優先権でのリソースの彼に対する割当てが非常に限られたものであるかもしれず，このような場合には優先権も役に立たないことがあります．ユーザがログインしようとするときに，システムが一杯だといわれることもあるでしょう．このような場合，もし彼が優先権をもつユーザで，これを主張すれば，システムは多分低い優先権のユーザをログアウトして，彼のために場所を作ってやるでしょう．ユーザは優先権の見地からグループに分類されています．これは，機械のリソースをそれを使っている共同体内で公平に分配するために必要であります．それ故，一つのグループに属する優先権をもつユーザが，他のグループに属する優先権をもたないユーザを追い出すとは限らないわけです．

マルチアクセス計算機システムの開発にはまだまだ時間がかかります．しかし世間ではコンピュータネットワークやコンピュータユティリティについてすでに語りはじめています．これに関して時々混乱が生じているように私は感じます．通信網というものと，通信網に接続されている計算機システムとは，極めて明確に区別しなければなりません．現在までのところ，私たちは元来電話と電信のために開発された通信網を使っていますが，これは実のところデータに適したものではありません．今後数年以内にはデータ通信にもっとずっと適した通信網が開発されることと思います．このような通信網では，接続の平均保留時間はずっと長いが，平均のデータ量はもっと低くなるでしょう．これらの通信網が安価に機能を提供してくれること，すなわち，現在は高価な機能をもっと安く供給してくれることを期待したいと思います．通信網の寿命と計算機システムの寿命との間には大きな開きがあります．計算機というものは5年で時代遅れになるのに対して，私がさきほどから話しておりますデータ通信網は，一たび建設されれば多分 50 年あるいはそれ以上も使われるものでしょう．

一つの計算機システムは一つの共同体にサービスし，それ特有の標準と操作上の約束をもっています．

異なるシステムはその設計者の異なる思想を，またそのシステムがサービスする共同体の異なる要求を反映するものでしょう．それにもかかわらず，異なるシステム間の通信は完全に可能だろうと私は思います．今や私たちは計算機に情報を文字の形式で貯えることを学びました．そして文字の形式であるからには，それは本質的には機械に無関係であり，一つのシステムから他のシステムへと送れるものであります．異なる計算機は文字を表現するのに異なるコードを使っているかもしれず，何らかの変換が必要かもしれません．しかし一たび情報が計算機内に入れば，一つの形式から他の形式への変換は，一つのアルゴリズムによって容易に遂行することができます．

私は，“ユーティリティ”という言葉が，その筋によって監督され供給されるところの公共ユティリティを暗に意味しており，この“ユティリティ”という言葉が計算機システムに適用されるのは間違いであると思います．通信網はたしかに公共ユティリティであり，公共サービスとして供給されねばならないものであります．しかし計算機システムはこの公共サービスのユーザと見做されるべきものであります．郵便の場合についての類似によって，このことがよく判ると思います，英国や米国では通信販売というものが非常に盛んでありまして，大ていのものは郵便で買うことができます．しかし通信販売事業が郵便を，それも単に郵便を使っているからといって，これは公共ユティリティではないか，などという人はいないでしょう．

現在，多くのタイムシェアリングシステムが稼動していますが，マルチアクセス計算システムというものはまだ極めて幼稚な状態にあることはお判りだと思います．数年のうちには，どんな大きさのものでも，あらゆる計算機がそれに接続されたいくつかのコンソールをもつようになり，たとえそれらが制御および監視の目的およびプログラムの開発の目的にだけ使われるとしても，いくつかのコンソールをもつようになると私は確信しております．今後開発しなければならないことは沢山ありますが，信頼性があり，かつ経済的な大きなタイムシェアリングシステムを私たちが持つようになるのはそんなに遠い将来ではないと私は堅く信じております．

有難うございました．

訳者謝辞：この翻訳は電気試験所電子計算機部加藤雄士氏が当日の録音テープによって記述された英文のテキストにより行なったものです．録音テープから英文を記述するという面倒な仕事を厭われなかった同氏に謝意を表する次第です．