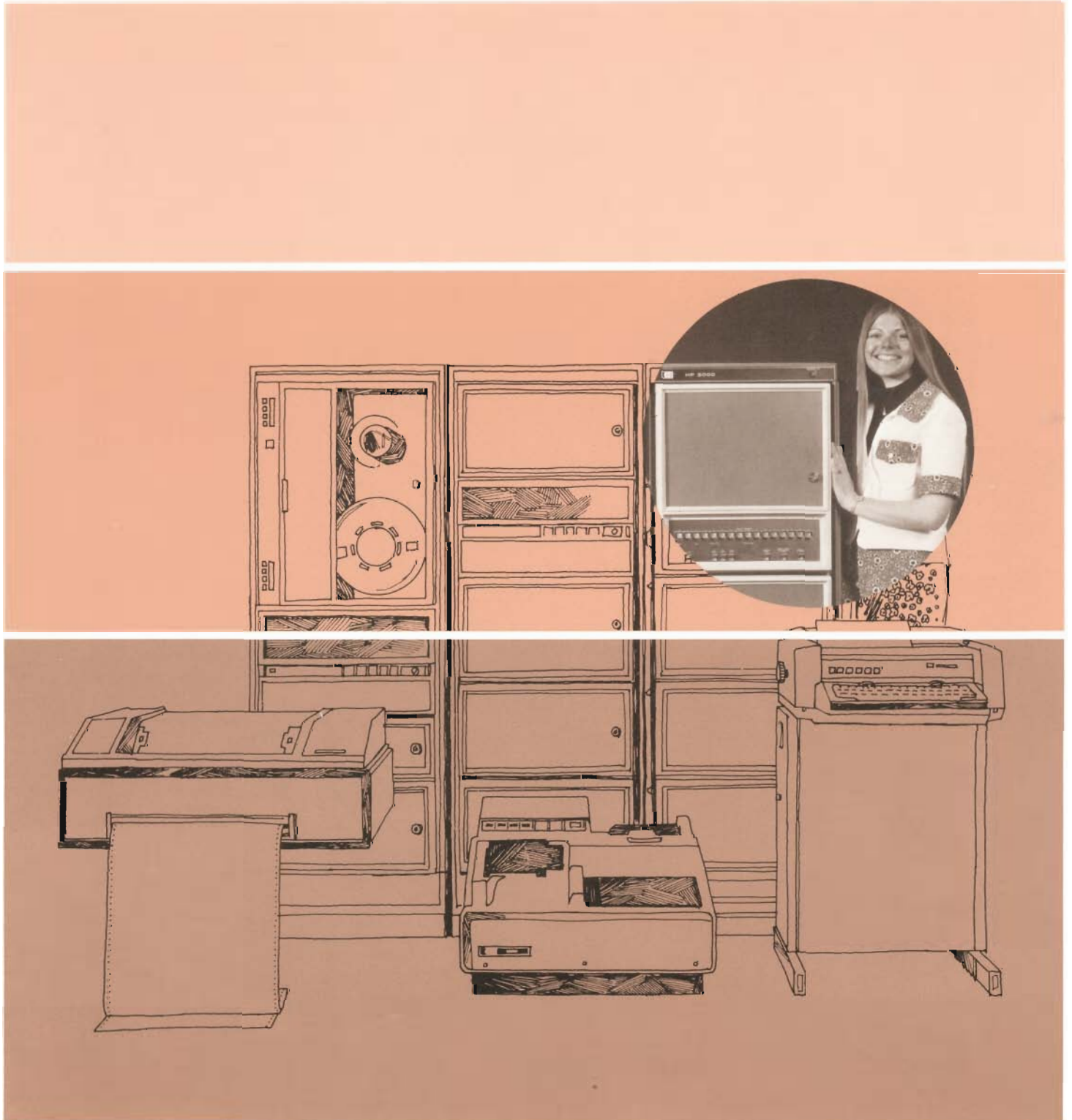


Focus In On
Advanced BASIC Timesharing
HP 3000 Models 100 and 200

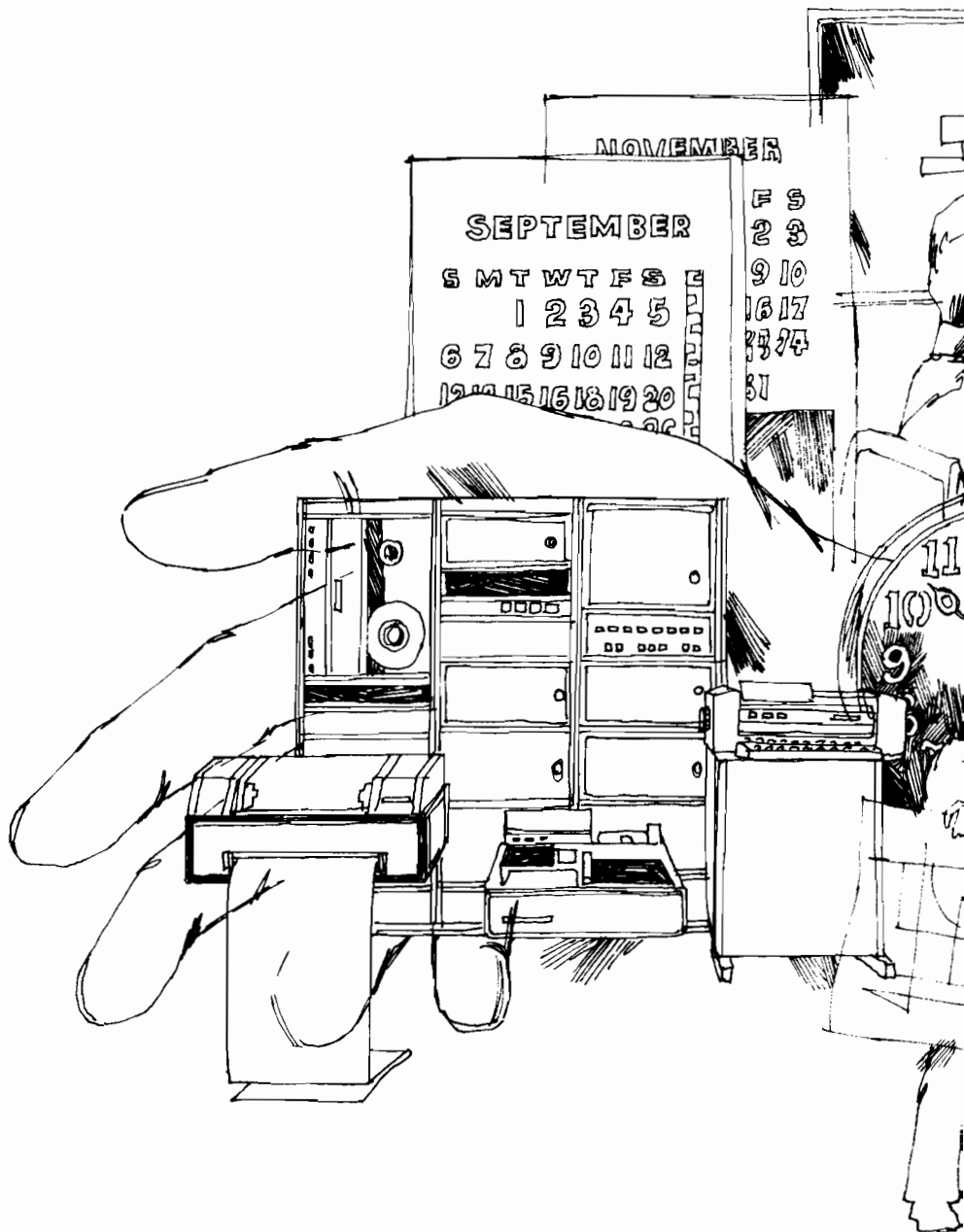


Let's take a close look at: YOU, TIMESHARING, AND HP

The key to handling your timesharing needs is at your fingertips, from HP! We know the right timesharing system saves you money. And, advanced BASIC language and system capabilities are a must for knowledgeable and innovative users like yourself. That's why we introduced the HP 3000 system, with the most powerful BASIC language ever.

The new HP 3000 Models 100 and 200 are the most capable minicomputer timesharing systems available, based on HP's years of experience providing general timesharing systems. Did you know HP has installed over 400 of these? All for different organizations, with differing needs. With the HP 3000, you get the best interactive BASIC offered on any minicomputer-based timesharing system. What you *also* get is capability to solve your most demanding applications; there's a full BASIC compiler that provides an average of ten-times more throughput than you would expect from a minicomputer system! FORTRAN, HP's own Systems Programming Language (SPL), COBOL, and IMAGE, a data base management system, are languages available to you. For communication with other systems, an IBM 2780/3780 emulator is also available. You can configure your system to provide interactive BASIC at terminals, and concurrent multilingual program compilation and execution in the traditional batch mode. Or, configure so terminal users may access compiled programs—even multiple languages!

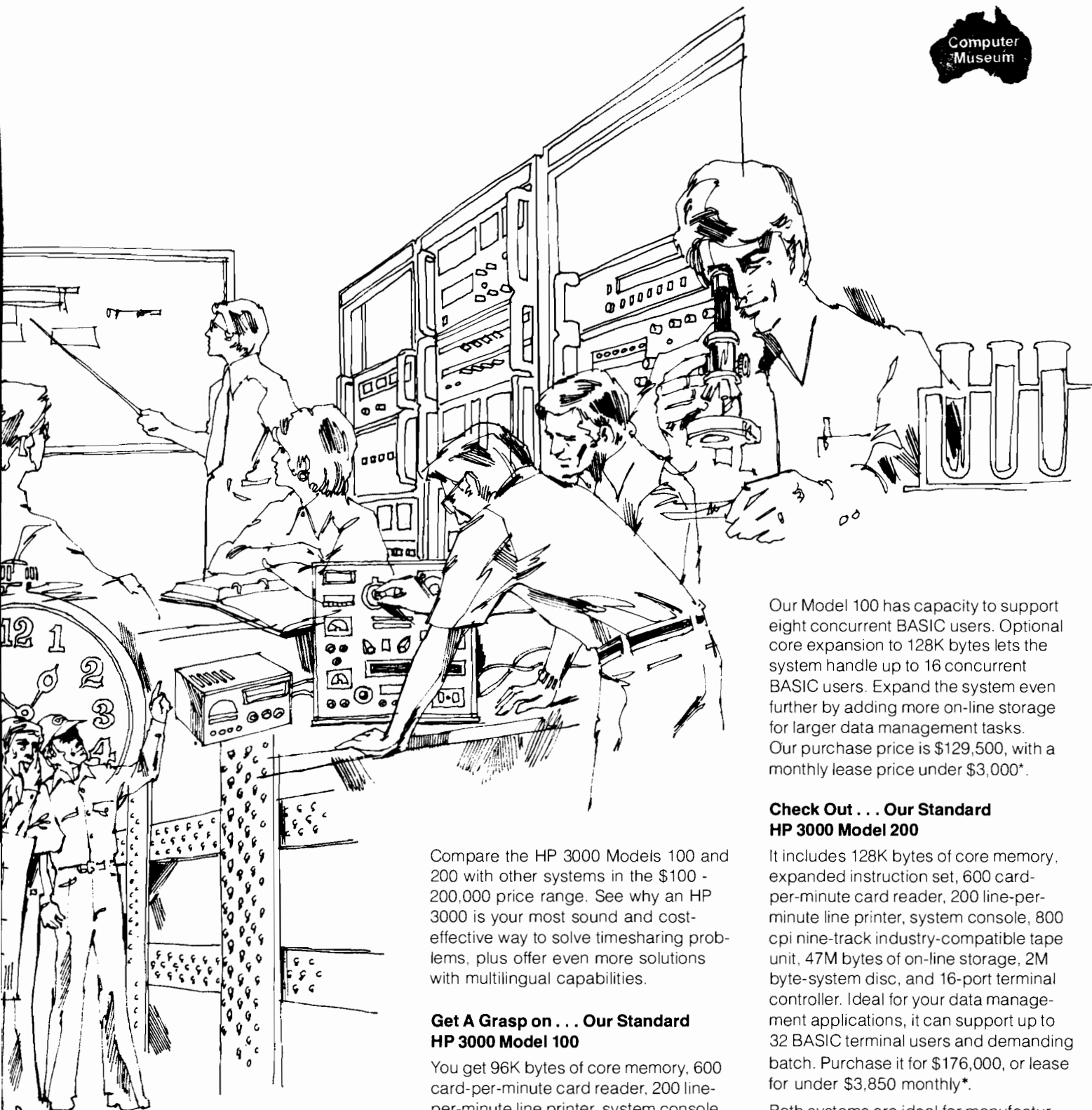
With the HP 3000 you can easily access system resources and peripherals. And, for demanding data base management tasks, you'll have the most advanced file capabilities on a mini-based computer system. The same file system services both time-share and batch, giving you quick data entry and retrieval, while additional data handling and reporting can be batched. *And, it's all within your reach . . .*



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Focus on state-of-the-art technology: YOUR MOST COST-EFFECTIVE INVESTMENT



Our Model 100 has capacity to support eight concurrent BASIC users. Optional core expansion to 128K bytes lets the system handle up to 16 concurrent BASIC users. Expand the system even further by adding more on-line storage for larger data management tasks. Our purchase price is \$129,500, with a monthly lease price under \$3,000*.

Check Out . . . Our Standard HP 3000 Model 200

It includes 128K bytes of core memory, expanded instruction set, 600 card-per-minute card reader, 200 line-per-minute line printer, system console, 800 cpi nine-track industry-compatible tape unit, 47M bytes of on-line storage, 2M byte-system disc, and 16-port terminal controller. Ideal for your data management applications, it can support up to 32 BASIC terminal users and demanding batch. Purchase it for \$176,000, or lease for under \$3,850 monthly*.

Both systems are ideal for manufacturing, engineering, finance, operations research, technical centers, and universities.

*based on five-year, full payout lease, not including maintenance.

Compare the HP 3000 Models 100 and 200 with other systems in the \$100 - 200,000 price range. See why an HP 3000 is your most sound and cost-effective way to solve timesharing problems, plus offer even more solutions with multilingual capabilities.

Get A Grasp on . . . Our Standard HP 3000 Model 100

You get 96K bytes of core memory, 600 card-per-minute card reader, 200 line-per-minute line printer, system console, 800 cpi nine-track industry-compatible magnetic tape unit, 9.8M bytes of on-line storage, and 16-port terminal controller for terminal connection (hardwired or via 103-type modem).

Examine the reasons why: HP 3000 — YOUR BEST TIMESHARING BUY

Knuckle down on your timesharing problems with our advanced BASIC language. You'll be handed:

- Character string operations to easily manipulate alphanumeric data for quick retrieval.
- PRINT USING AND IMAGE statements for formatted output. Program reports easily.
- Matrix operations like reading, inputting, printing, initialization, and copying. Get the right data at the right time.
- Ability to call compiled subroutines written in FORTRAN, COBOL, or SPL. Get extra programming flexibility for the extraordinary job.
- Concurrent support of multiple data types (integer, real, long, and complex) for efficient data storage.
- Extensive commands for program debugging and optimization; features like tracing variable paths and changes, displaying dynamic nesting structures, and summarizing statement usage. The result? Faster applications development and optimal system use.
- Up to 64K-character programs for users; lessens the need for chaining, for simpler programming.

Grab On To . . . Advanced resource access to I/O devices like line printer, card reader, and magnetic tape unit. Get an industry-compatible magnetic tape unit with each system, and have efficient program/data backup and interchange with other systems.

Put Your Finger On . . . Maximum system utilization with concurrent multilingual batch. Time not required for interactive support is automatically allocated to batch jobs; no need to

shut-down and restart the system for batch processing!

Get . . . Autospeed detect and type 103 modem capability with a standard 16-port terminal controller. Only one phone number serves all terminal types. 202 type modem capability is optional.

Get A Feel For . . . The best multiprogramming system architecture!

Modular, bus-oriented, with 5.7 million byte-per-second bus transfer speed, and separate I/O processing for concurrent I/O and CPU processor. State-of-the-art power supplies minimize power needs. Hardware-implemented stack permits fast context switching, and efficient subroutine calls and memory use. A powerful 32-bit LSI bipolar ROM microprocessor with a microprogrammed instruction set efficiently serves a high throughput, multiprogrammed-timesharing environment.

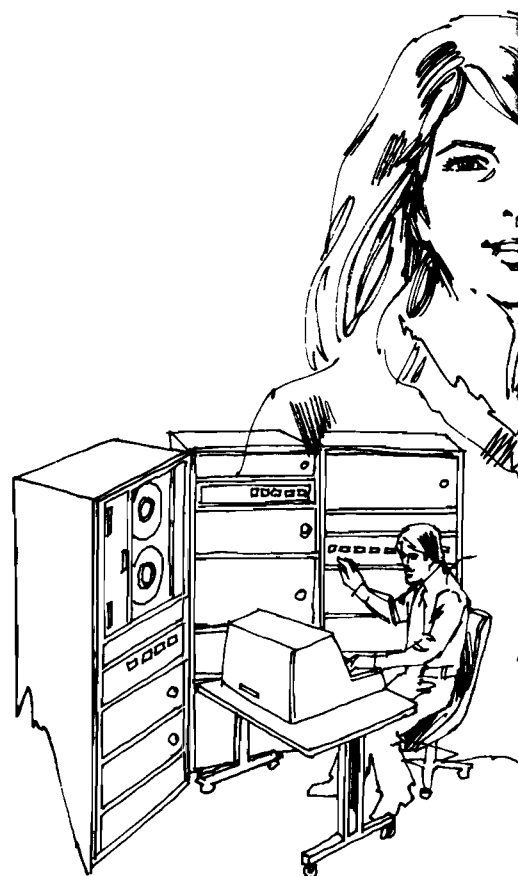
Handle . . . System status displaying, accounting, and usage limits with accounting and logging facilities.

Touch Base With . . . The most powerful file system available on any minicomputer system! Comprehensive data-handling capabilities plus data security and integrity efficiently implement extensive on-line applications. This file system provides:

- Three file types: formatted, ASCII, and binary. Combination provides easy-to-use and efficient files . . . powerful data management capability for novices or experts.
- Static security at several levels, plus dynamic file security, assuring you that sensitive data is kept confidential.
- Up to 16 files accessible to each user at once. Programmatic file opening and closing allows access to an unlimited file number. Get on-line file size to 47 million characters or expand on-line storage to over one billion characters . . . capabilities for now and the future!

- Common files for interactive users and batch jobs, for operations flexibility; data entry on-line during normal operation hours, and high speed, powerful batch sort/merge during off-hours using SORT/3000.
- Efficient system backup on magnetic tape for data integrity. Back-up five million bytes in less than two minutes!

Get . . . Minimum operations cost with straightforward system management. The powerful operating system efficiently allocates CPU, memory, disc, and other system resources for maximum throughput and total work output, built on the most-advanced minicomputer architecture.



DUMP

recap

PROCESS

memory system

group

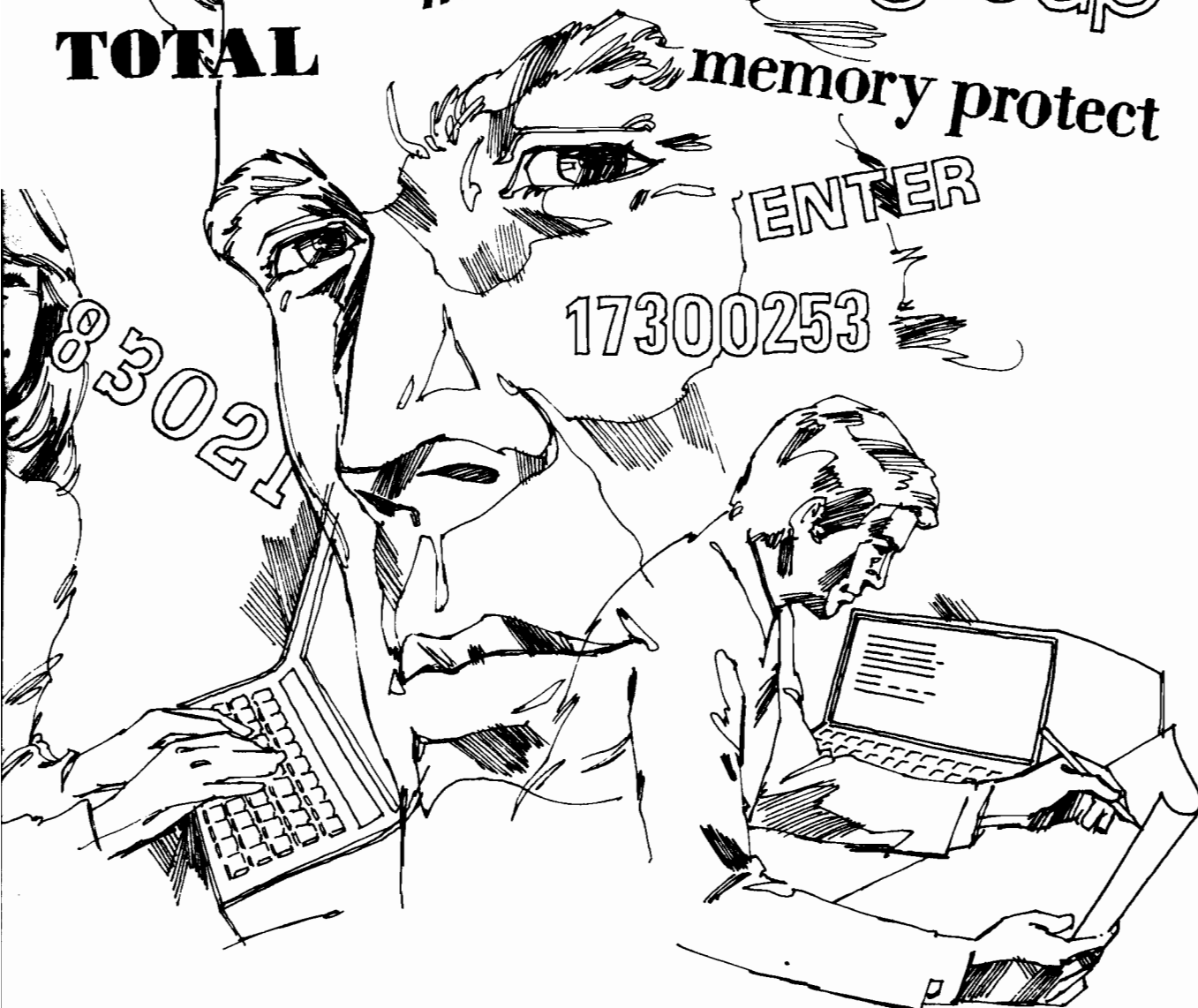
TOTAL

memory protect

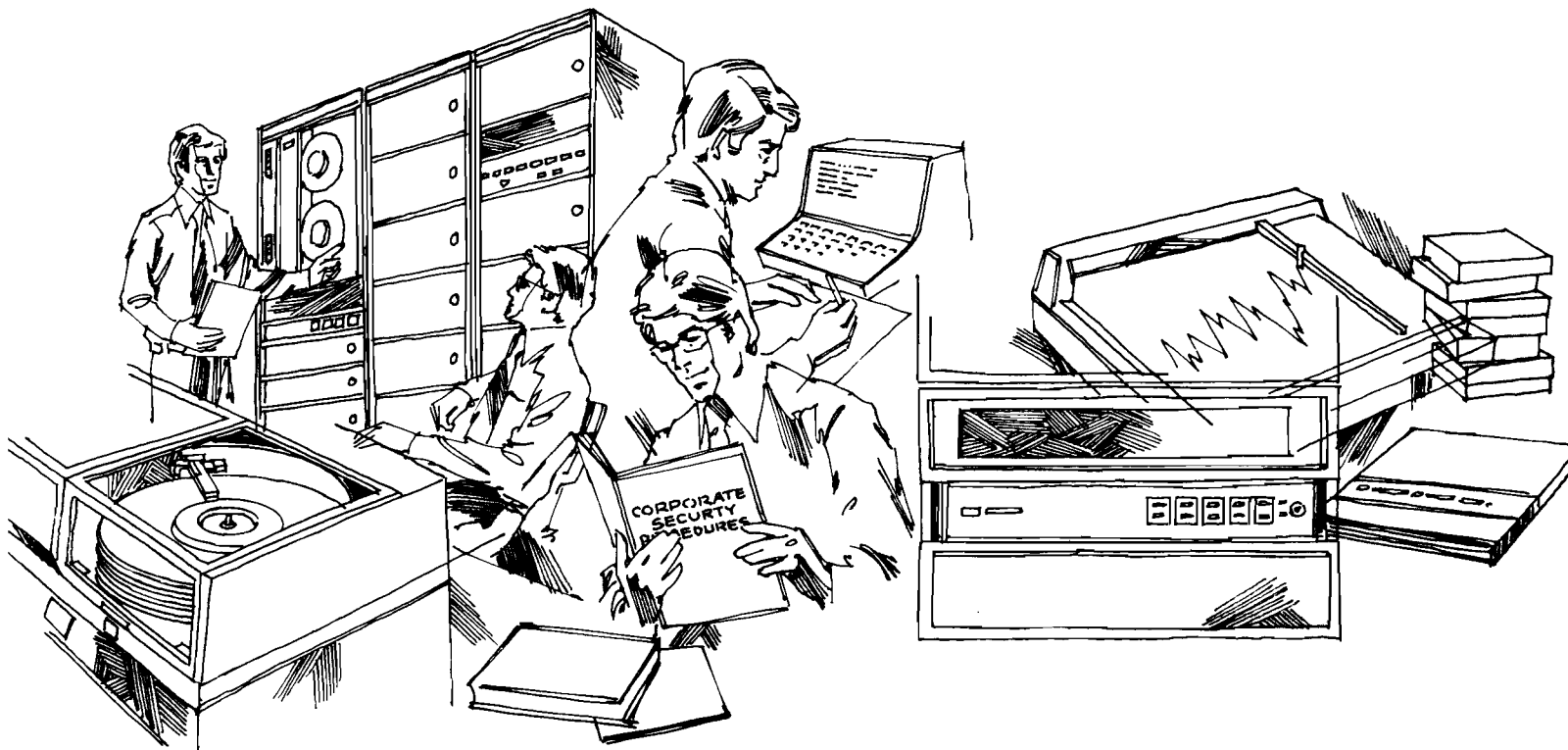
ENTER

17300253

85021



Take a timely look at: UNIQUE DATA MANAGEMENT



HP offers data handling normally found with larger data processing systems. Put sensitive data on the system, confident that it is secure from other users. Want to share data? You can make it available to users with access restrictions. Advanced features include: altering file specifications at run time, programmatically referencing files, and I/O resource access through file system.

Use these three file types:

- **FORMATTED** files provide advanced, easy-to-use capabilities. Created automatically by BASIC, they insure you data integrity by using format words for data items.
- **ASCII** files, created via the operating system's file commands, are accessed using print and input statements.
- **BINARY** files created and accessed like ASCII files, store data items as binary words, giving you optimal storage.

Files are easily created, updated, ex-

tended, or purged from the user terminal, or totally maintained under program control. Open up to 16 files concurrently. Over 16 can actually be referenced by a program, thanks to programmatic reference. This means the system can effectively support large data base applications.

BASIC-Formatted Files For Every User.

BASIC-formatted files include serial and direct access, using PRINT and READ commands. Two useful statements are ADVANCE and UPDATE. ADVANCE lets you skip forward or reverse past items in a record without reading them; UPDATE lets you update a single item in a record. Both statements simplify data retrieval and updating. Basic-formatted files are created using a BASIC CREATE command.

The DUMP command lets you print file contents at the terminal. Or specify other output devices, like the line printer.

ASCII Files For Greater Applications Flexibility.

Access them via PRINT and READ commands. You get serial access for variable length record files; serial *and* direct access for fixed length record files.

Reading data from an ASCII file with special characters is simplified by a LINPUT statement. A record's contents are read into a string variable; all characters are read into the string variable. Easily extract individual data items with string handling commands.

Binary Files For Maximum Data Storage Efficiency.

You get PRINT and READ commands, and serial access for variable length record files; and, serial plus direct access for fixed length record files.

Need to access variables from a binary record? Then, specify substrings instead of the full string variable, and extract part of a data item easily.

Matrix print and read statements provide file array operations. Both serial and direct reads are provided, letting you work easily with large arrays. When files are created, they can be divided into equal-sized extents. You only need to allocate the first extent at creation, so the disc file can grow as required. Consequently, you can anticipate and handle applications expansion. Also, instead of the default buffer size, you may specify the size of the buffer used by the I/O system. This means maximum individual program I/O efficiency!

Comprehensive File Security For Multi-Level Access Control.

You need data and program file security. Here's how we provide an extra security measure.

All files belong to a group; groups belong to a higher entity — an account. File security provisions exist at each level; total file security results from total security of these levels. Normal (default) security allows users all access modes to files within the user's log-on group. There is a systems library for everyone's use, plus a general one for each account.

With account or group management capability, you can easily alter these normal provisions for the group.

The user controls the security of each file he creates, but must use a security at least equal to the group's and account's. So, a file can benefit from group security specifications, although not explicitly protected at the file level. The group/account security mechanism provides access checks; but there's more . . .

Once you open the file, dynamic security allows these access specifications:

- No other user may access the file.
- Other users may read and write on the file.
- Other users may read, but not write on the file.
- Other users may read the file or read and write on the file, yet, the program can lock the file.

The dynamic locking capability is important when on-line disc files must be accessible to many users at once.

File lockword capability completes the extensive file security. BASIC programs can be saved for run-only operation.

Straightforward On-Line Peripheral Access.

Use the file system for I/O to card readers, line printers, magnetic tape units, and other peripherals. Read-in data and/or programs easily from a card reader. Or, direct output to a high speed line printer, or magnetic tape unit.

Just establish a file reference using the FILE command. Programmatic reference to the I/O device is then via the file name. Change the I/O device, *without changing the program*, by altering the device reference at run time.

A program may call for data input from a card reader or magnetic tape. Your program, then, references a file that we'll call DATIN; to input data from the card reader, the file command entered at run time specifies DEV=CARD. For input from magnetic tape, the device specification is simply changed to DEV=TAPE.

Addressing on-line I/O peripherals via the file system also eases programming. System-supplied specifications for "files" referencing I/O devices avoid the need for complex I/O device specifications—yet, you can alter these specifications for unique I/O requirements.

On-line access to I/O peripherals is also a means of file storage and/or backup on magnetic tape. STORE and RESTORE commands are used to store and restore files (either data or programs to tape). The industry-standard tape unit can backup a five million-byte file in less than two minutes. File information, including security data, is automatically stored on the tape. Because this security data is included on tape, only the user with proper security information can restore the file from the tape. Magnetic tape is inexpensive for storing libraries, data files, and sensitive data, off line, besides providing effective interchanging with other computer systems.

Disc Files.

The HP 3000 Model 100 uses two efficient and dependable moving-head cartridge disc units. Each drive stores 4.9 million bytes of information; you can add drives to the controller.

Increase on-line storage capacity for the system by adding 47 million-byte moving head disc units.

The HP 3000 Model 200 includes a fixed-head disc with 2 million bytes of storage capacity. With average access time of 8.5 milliseconds the disc assures you maximum system response and performance. You can expand it to 4 million bytes.

Our HP 3000 Model 200 also uses a high-capacity, 47 million-byte fast access disc. The controller supports up to eight of these discs. You can expand to 376 million bytes with the supplied interface. And, individual disc files may be as large as 47 million bytes. More interfaces mean expansion past one billion bytes!

Put your finger on: MULTILINGUAL AND BATCH CAPABILITIES

To give you maximum system utilization and throughput, HP 3000 provides a concurrent multilingual batch capability. Time that interactive users do not need is automatically allocated to the batch queue which supports multiple batch jobs. This means full system utilization at all times!

The HP 3000 is the only mini-based timesharing system to offer advanced BASIC, with features that effectively solve today's demanding problems . . . and batch access. With batch access, the system can efficiently solve a wider range of applications, whether they require immediate response or scheduled reports.

These multilingual service capabilities are all available to you: BASIC/3000, FORTRAN/3000, SPL/3000, TRACE/3000 (a programmatic debugging system), SORT/3000 (for sorting and/or merging files), and a utility for general file copying and translation. Get COBOL/3000, and IMAGE/3000 Data Management System, as well as, 2780/3780 emulation. And, expand the terminal user access to include compiled programs and these multilingual services!

FORTRAN/3000. Based on ANSI STANDARD FORTRAN (X3.9-1966), its extensions expand language capabilities and power. Features include accept and display statements, character variables and arrays, bit and byte manipulation, and up to 255 dimensions in arrays. FORTRAN's most significant feature is its efficient compilation and execution time.

The advanced compiler produces efficient object code because the HP 3000 has the most-advanced minicomputer instruction set; for example, in loop control instructions, only one instruction is needed to modify a variable, test against limit, and branch. Small overall program size also results from dynamic storage allocation—less memory is generally required for program execution. And, you get short compilation times, expected only from large-scale processors.

SPL/3000. The HP 3000 Systems Programming Language (SPL) is both high-level and machine-dependent. It offers you a high-level language similar to ALGOL for usual coding sequences. At this level you need no in-depth knowledge about the system architecture and operating system. With increased knowledge, you can exert control over any system machine-dependent functions.

Frequently-used, or critical, routines implement with SPL—and call from BASIC or any of the other languages. If you are a knowledgeable programmer, implement library routines, and make them available to other users to benefit from increased execution speed.

EDIT/3000. The Text Editor lets a user modify text and program files.

IMAGE/3000. A complete Data Base Management System with unique capabilities for the toughest applications. You can: describe data base structures like items, security, data sets, data set relationships, and storage needed; use a language to define file structures; create data bases; and access and maintain data. Utilities let you easily restore data bases from magnetic tape.

QUERY/3000. Lets you easily locate, report, and update base values within an IMAGE/3000 data base.

SORT/3000. Sort and/or merge multiple data files of sequential records into one sequential file. Multiple sort keys can be contiguous, separated, or overlapped, giving you sort/merge capabilities expected only on large-scale processing systems.

COBOL/3000. It conforms to the highest level of federal government standard COBOL and has the added capability of inter-program communications. It provides you with COBOL's advantages plus programmer-controlled object code segmentation for more efficiency.

COMMON SYSTEMS COMMANDS.

A Common System Command Language for terminal and/or batch access means you can easily switch access methods, using the best one for your problem. Another advantage is that the same files are used for both terminal and batch access. For example, interactive BASIC users can input to daily operational files; during non-peak hours, the files can be manipulated by batch access for sorting or merging via SORT/3000, or additionally processed for data management, using FORTRAN or COBOL and IMAGE.



Investigate our stack architecture for: EFFECTIVE SYSTEM MANAGEMENT

Efficient System Resource Use.

Its basis is the Multiprogrammed Executive/Timesharing operating system. This software system means maximum CPU use, and interactive terminal user service. Time not required for interactive terminal work is automatically used for batch background work. Your system continuously achieves peak throughput.

System overhead is minimized, because the hardware provides many instructions for system functions. An example is the linked list search instruction, letting the operating system quickly determine how much free memory is available. You get an efficient memory management scheme.

User Resource Access Management.

System users are associated with different groups, which belong to higher, "account" entities. The system manager creates and adds accounts. Within each

account, a manager establishes or adds groups and users. Users, groups, or accounts can be added from any terminal with proper identification.

Set limits on file space, central processor time, and connect time when creating these accounts or groups. Specify passwords and system capability attributes for users, groups, and accounts. For example, a user can be assigned to either batch access, interactive (terminal) access, or both.

Character names can identify users, groups, and accounts, so relating to the user's function is easier. At log-on, the user only specifies his user and account "ID"; each user is automatically associated with a "home" group. A user may log-on outside of his home group, if he knows the group "ID". Passwords may be specified at user, group, and account levels for tight system access security.

A system-wide public account facilitates information and program sharing, and there's a public group for each one. The account manager designates one librarian to manage the public group within the account!

MPE 3000 accounts for system resources used by each account and group; these include total permanent file space, CPU time, and terminal connect time. Extract and display accounting information for each user, group, and account with a REPORT command with appropriate access clearance.

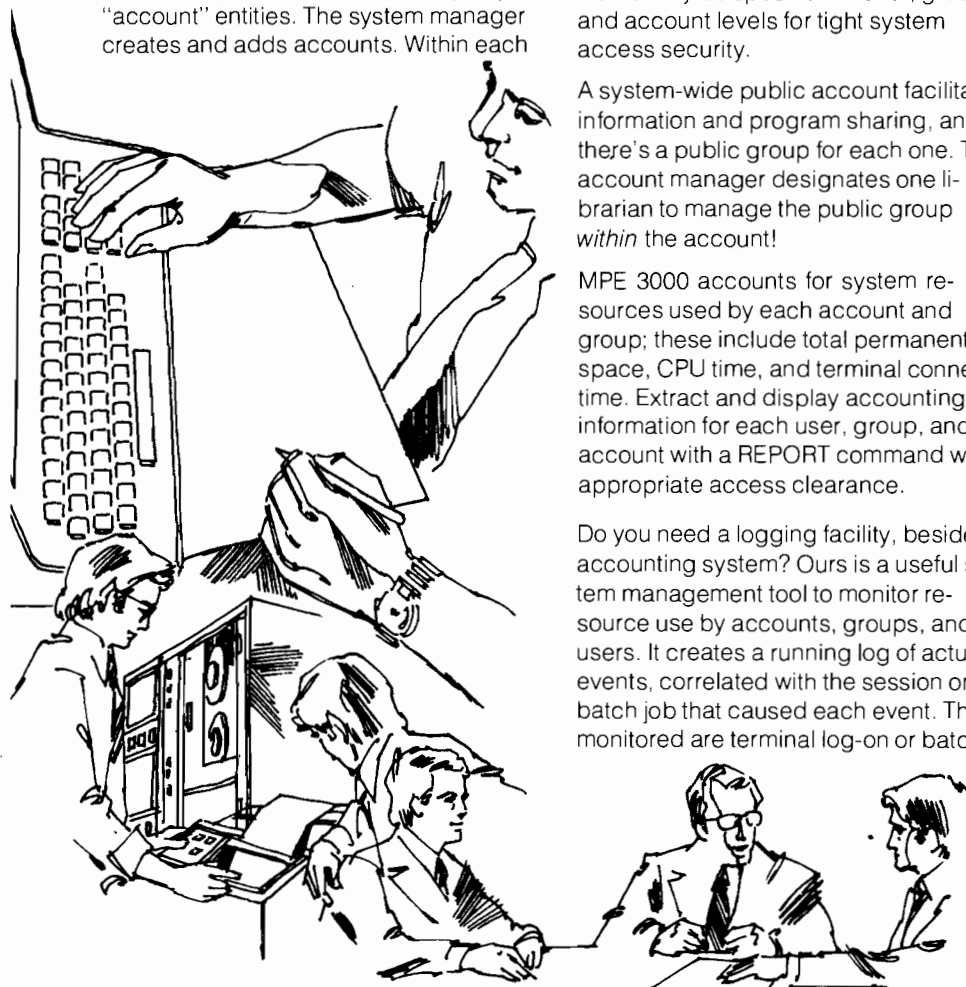
Do you need a logging facility, besides accounting system? Ours is a useful system management tool to monitor resource use by accounts, groups, and users. It creates a running log of actual events, correlated with the session or batch job that caused each event. Those monitored are terminal log-on or batch

job initiation and terminations, file closing, and system information on system shutdown and reload. The logging facility is a great tool to overview system usage and monitor various dynamic resources. Logging, itself, and events to be logged, are optional; you can activate them during system configuration.

System Integrity. To restore on-line files to a known state, we give you straightforward methods of dumping system files and restarting the system, saving you time. Use standard high speed magnetic tape I/O. To back up a system, the system manager utilizes a SYSDUMP command, typically when no other users are on the system (SYSDUMP is valid with users on the system; however, files in use with write access are not dumped).

You can make system configuration changes when a system is backed up, like maximum seconds allowed for logging-on and I/O changes. Especially important is dump date specification. A dump date of "zero" stores the system, accounting structure, and all files. With a specified dump date, the system, accounting structure, and any files changed on or after that date are stored. On the weekend, then, when more time is available, you can make a complete dump. During the week, only changed files (as on a daily basis) need to be dumped, shortening your required dump time.

Some applications require file backup for certain files or accounts more frequently. You do this easily with the STORE command. Files to be sorted can be defined, or just designated as all the files associated with a group or an account. For large data bases, STORE supports multi-reel files. Use the STORE command on-line (and, the RESTORE command for reading files back in,) while still serving other users. These backup methods offer the operating flexibility you need for a variety of operational requirements, plus lots of file integrity!



We proudly point out our: NEW ARCHITECTURAL APPROACHES

Microprocessor.

Here's our way of insuring the flexibility, speed, and growth expected from a state-of-the-art system.

A 32-bit, LSI bipolar ROM-based microprocessor forms the heart of the HP 3000 system. This high technology, high speed unit is capable of over 5,700,000 operations-per-second. The microprocessor is the basis for 176 firmware instructions, making our HP 3000 the most powerful computer in its class. And, due to the microprocessor design flexibility, future instruction-set expansion is close at hand.

The 176 firmware instructions, (all single-word formats except the two-per-word stack operations), offer maximum computation speed and code compression. Tailored to the compilers and operating system, the instruction set lets compilers generate highly efficient code—up to 40% less than similar class systems. Your operating system overhead is minimal. Microinstructions are coded for functions like interrupt-handling, seeing if a code segment is resident in core or on disc, or determining the available free memory. And, computational speed for arithmetic operations (add, subtract, multiply, and divide) is implemented for 16- and 32-bit fixed point quantities, and 32- and 48-bit floating point quantities!

What makes HP 3000 microprocessing fast and efficient? "Pipelining" does; each microinstruction is divided into two phases—an operand fetch phase and an execution phase. These overlap, so the execution phase of one microinstruction takes place when the operand fetch phase of the next instruction does. ROM microinstructions are thus "pipelined" at the rate of 1-ROM-word per cycle, increasing throughput and speed.

Modular Organization Built Around A Central Data Bus.

Here's HP's assurance of expandability and efficient multiprogramming operation.

The HP 3000 uses independent hardware modules organized around a 175-nanosecond data bus—truly high speed and state-of-the-art! Parallel transfers (16 bits) flow between these multiple, independent, asynchronous modules during each bus cycle. And, since central data bus communication is synchronized with the bus clock, a bus cycle allocation scheme eliminates the common "handshaking" operation mode. The data bus then operates at the highest possible rate, resulting in unmatched system throughput!

CPU and I/O processor separation means the HP 3000 has concurrent CPU and I/O processing. The result? Maximum CPU utilization for computation—no loss of CPU cycles to DMA transfers or I/O handling and buffering! Multiple memory modules with up to 64K bytes each are provided. Access them independently or interleaved to increase the

effective data rates in and out of main storage. Asynchronous terminal controller modules with autospeed sensing, provide simultaneous access by 16 remote or local terminal ports. A 16-port terminal controller is an integral part of this system, built especially for multiprogrammed timesharing.

The I/O processor, independent of the CPU, controls the 880 kilobyte-per-second multiplexer channel, and provides access to the many peripheral devices supported on our HP 3000. If I/O data rates exceed the multiplexer channel capacity, you can add 2.8 megabyte-per-second high speed selector channels. And, selector channels connect directly to the central data bus, so, data transfers are concurrent with devices on the multiplexer channels.

The HP 3000's asynchronous I/O scheme means flexibility to handle both slow unit record devices and high-speed tapes and discs concurrently. This is significant independence for a system built to adapt and take advantage of rapidly changing technology.

Examine that critical resource: EFFICIENT MEMORY MANAGEMENT

Stack Architecture.

Get high level system performance expected only from very large computer systems.

An HP 3000 program operates in a domain consisting of a data stack (a linear data list whose last element is in prime computing position), an instruction (or code) segment, and a global data area. We pack our versatile stack instructions two-per-word for efficient high-density coding. Registers define stack locations (top of stack, bottom, subroutine entry marker, and so on), so time is not wasted loading and unloading stack registers during execution. The top four stack locations are implemented as hardware for even more speed. 90% of the CPU operations use this top-of-the-stack hardware, resulting in high speed CPU processing.

The stack structure is efficient for parameter passing, dynamic allocation of temporary storage, efficient arithmetic expression evaluation, and recursive sub-routine or procedure calls. The stack also enables rapid context switching—21 microseconds to establish new environments upon interrupt. Our HP 3000 has all stack features, including checks for over- and underflow, implemented in hardware. This makes your multiprogrammed timesharing environment even more efficient.

Your most critical system resource, memory, should insure: a) multiple users are in memory simultaneously, preventing needless system work, b) programs larger than available memory can be handled without user intervention, c) programs resident in memory only utilize the space they actually need, d) minimal memory is fragmented, e) swapping of user programs between disc and memory is minimized. Our HP 3000 uniquely meets these requirements. Here's how:

Separation Of Code And Data.

All HP 3000 programs are automatically separated into static code and dynamic data segments. The result? All program code is automatically reentrant, and can be exited and entered any time during execution. And, all code on the HP 3000 is recursive (program code calling itself), a must for efficient compilers and system software. These unique qualities mean system and user code on the HP 3000 can be shared among several users in memory simultaneously. The unique element is each user's own data space. So, the total memory needed for program execution is far less than if multiple copies of the same program were required to reside in memory.

And, since user and system code is never modified, it need never be swapped out of memory. Code is automatically overlaid, dramatically reducing system overhead and swapping time; this insures you more system resources for more users.

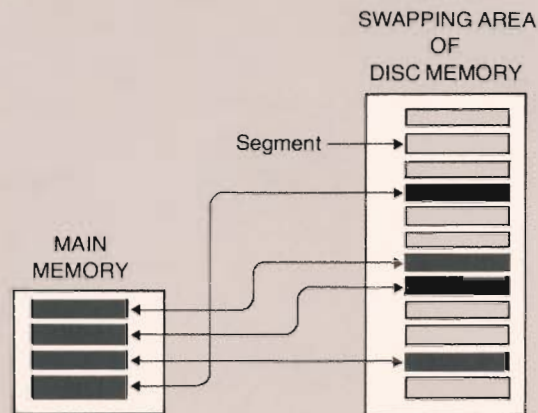
Virtual memory.

The HP 3000 gets its virtual memory power and flexibility from variable length code segmentation controllable by the batch user. Our hardware and software automatically handles all code and data segment management of the compiled programs on the system. Consequently, you have main storage for designing programs, without needing planned overlay or dynamic management techniques. If a program is too large for main memory, the user simply segments the program along logical boundaries. From there, the operating system takes over. All program code and data segments are addressed relative to hardware registers, for simple segment relocatability.

This relocatability is used by the virtual memory manager to optimize main memory use and prevent memory checkerboarding. And, system hardware registers restrict users from accessing memory outside of their environment. This protects program and data integrity of individual users *and* the operating system. Main storage is shared dynamically among every active user code and data segment. And, programs get the main storage they need when they need it—but only *what* they need! This virtual memory technique adapts to the changing demands of the HP 3000 multiprogrammed timesharing activity.

At your fingertips, from HP!

Virtual Memory



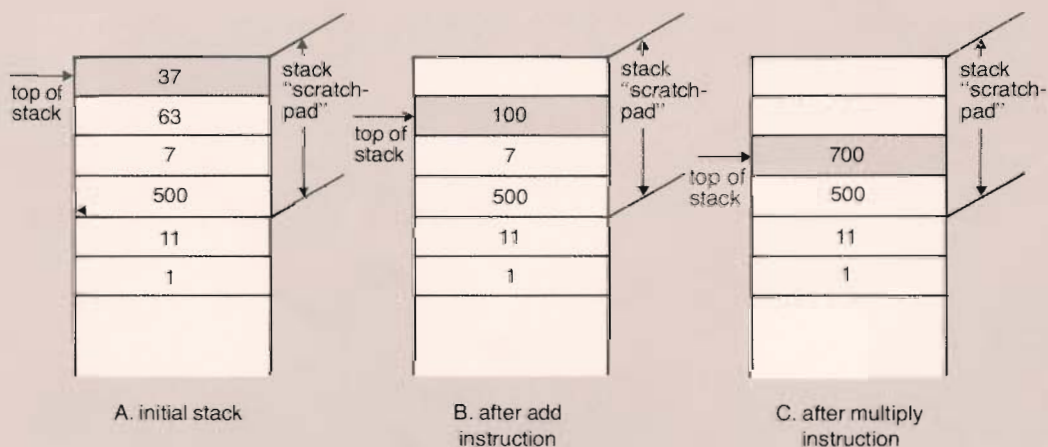
Data Stack Operation

Our unique stack structure consists of a linear data list, whose last element is in prime position for computational operations. Here's an example. Figure A shows a stack. The ADD instruction uses the top two stack locations, and places the result on the "top" of the stack (Figure B). The next instruction, MULTIPLY, multiplies these locations and again places the result on the "top". (Figure C).

You need not give an operand address in the instruction word since the "top" stack is its explicit address. So, the instruction simply is "ADD" or "MULTIPLY", in this case. Your benefit? Code compression—the ADD and MULTIPLY in this example requires only one instruction. And,

because the HP 3000 reduces the number of memory fetches for instruction words, speed increases. Temporary storage of intermediate results is also provided—automatically! For example, once the parameters 63 and 37 (at B) have been added, they are no longer needed, so discarded. The answer, substituted on the top of the stack, is automatically in position (adjacent to 7) to multiply. The result? No need to provide a dedicated location to save the temporary quantity 100 (or any of the other intermediate results).

And, you use a stack "scratchpad" for the computations, decreasing CPU processing time.



Look for sales, service, and support . . . AS CLOSE AS YOUR LOCAL HP REPRESENTATIVE!



With 172 service locations in 65 countries, HP lends a supporting hand where you need it, when you need it.

HP 3000 Series Timesharing systems are available for lease or purchase. The lease plan features:

- Periods ranging from one to five years.
- Provision for upgrading equipment.
- Lease Renewal — three added one year terms at the conclusion of the base period.
- Purchase Option — at a fixed percentage of the contract or fair market value.
- Early conversion to purchase.
- Nine month termination option.

Whether you choose lease or purchase, you will receive our HP service plan, as well.

Customer Assistance Agreement

This agreement provides for:

- Maximum system reliability through regularly scheduled preventive maintenance.
- Quick response in emergency situations.
- Insurance against bills for repair of high cost items.
- Budgeting simplicity through known annual cost.
- System documentation maintenance.
- Fewer administrative problems — each and every maintenance incidence does not require a purchase order.

Training

Select an HP training course at a nominal cost. Qualified instructors offer these courses regularly at five training centers around the globe. Get first-hand classroom and workshop knowledge, ranging from a few days to two weeks in length.

Documentation

You receive complete documentation, including software user manuals and operator guides, with each system.

Let HP handle your timesharing needs today!



Sales and service from 172 offices in 65 countries.
1501 Page Mill Road, Palo Alto, California 94304

Printed in U.S.A. 9/74 5952-4590 D