Multics Technical Bulletin MTB-424

To: Distribution

From: Steve Webber

Date: 08/28/79

Subject: Multics on ADP -- Software Release 10.3 EPS-1

Attached is a draft of the Software EPS-1 for Multics ADP. Several assumptions are made that may not be valid. These are:

- There will not be an absolute requirement for object level compatibility of user programs. This deviation from the standard compatibility constraint is not limited to the removal of read-alter-rewrite but also includes formats of bit offsets as stored in pointers; changes to IDCW formats; and clock reading.

- The first customer ship (FCS) release of ADP will be 3Q82.

- The performance of Multics on ADP for first customer ship is not important to worry about (since we can not compare it to nonexistent data on the current hardware it is silly to project a required comparative performance level).

- The simultaneous execution of the Functional Test System (FTS) (on-line T&D running under its own operating system in the mainframe) and Multics under control of some form of hypervisor is not necessary for first customer ship (i.e., the hypervisor is not part of the first system shipped).

INTRODUCTION

This EPS describes the major changes planned for the Multics software so that it will run on the ADP hardware. There are unavoidable incompatibilities that will result from the current plan but it is assumed that these are acceptable. The incompatibilities are occasionally reflected to user programs. It is thought that this is not a critical problem due to the nearly total use of high-level languages by Multics users. Tools will be provided to aid users in preparing for ADP.

RELATION OF ADP MULTICS TO OTHER MULTICS RELEASES

The following diagram illustrates the currently planned Multics software releases and the hardware on which they are to run.

	1979		1980		1981	1982
DPS	MR7.0	MR7.0B MR8.0	MR8.5	MR9.0	MR10.0	
DPS-E			MR8.5	MR9.0	MR10.0	
600S			MR8.5	MR9.0	MR10.0	
ADP				(MR9.3)		MR10.3

Table 1.
Current Release Plan

The software for the first release will be based on the last major non-ADP release before the ADP release. The initial ADP software will be based on MR9.0, but by the time of first customer ship all MR10.0 enhancements, features, and bug fixes will also be included. Hence, from the customer point of view release MR10.3 will be based on MR10.0.

In parallel with the ADP software effort Multics release MR8.5 will be developed. This release will run on any of the Multics hardware bases, including DPS-E and 600s.

RELATION TO NOS (NEW OPERATING SYSTEM)

The software changes as described in this EPS-1 are not those required for NOS. The changes necessary to run NOS (based on Multics) on ADP will be included in a separate document and answered by a separate EPS-1.

COMPATIBILITY CONSTRAINTS

The first release of Multics on ADP will include several incompatibilities with respect to previous releases of the software. These fall into the following major categories:

- user program incompatibilities
- operational changes
- input/output

User Program Incompatibilities

Multics (object-level) user programs may not run on ADP for several reasons. If this is the case, nearly all problems can be solved by recompiling. The following problems, however, will require source level changes (in addition to recompiling):

- use of read-alter-rewrite for locking when the opcode used is not one of the five such that preserve this feature on ADP,
- reading the calendar clock in assembly language (the format returned by the RCCL instruction is different),
- use of certain features in I/O channel programs, and
- any program (tool, debugging aid, etc.) that understands certain privileged register formats that are perpetuated to the outer ring. The registers/data of interest include:

  history registers
  PTWs, SDWs (not a serious problem)
  SCU data

Object-level incompatibilities (forcing a user to recompile or run a conversion program) include:

- ITS and ITP bit formats

Operational Changes

There will be many operational changes required for ADP. These will primarily fall into the areas of system (BOS and Multics) bootload and initialization, switch settings (from real to virtual, i.e., no more real switches), system console usage, and the various interfaces to the SSF in general.

Operators of ADP systems will still be required to do the same basic functions, but many of the specifics of how these functions are done will be different.

Input/Output

The ADP I/O hardware central (IOX) is different from the IOM. Although the peripherals supported will initially be the same as on pre-ADP systems, the details of how the programmer talks to the IOX and how the IOX talks to the peripherals are different. The programmer interface is largely upward compatible at the user-ring level, but a few minor changes may be required. The simplest and most reliable tract is to look at all I/O initiating software in the system and correct it as appropriate.

The incorporation of a centralized tape I/O package (tape_ioi_) to replace tape_nstd_ would make conversions slightly easier, as the new interfaces would be designed appropriately.

CONSTRAINTS ON SEPARATELY-PRICED SOFTWARE

There are no plans to provide any software assist in Multics to support enforcement of (or detect violations to) the use of separately-priced software.

IMPLEMENTATION ISSUES

There are four areas of concern with respect to how we will implement the ADP software. These are:

- multiple versions (non-ADP and ADP) of the system,
- tracking of bugs and merging of systems,
- the logistics of hardware availability, and
- plans for conversion software and scenarios..

Multiple Versions of Multics

We will have a single version of the software for as much of the system as is reasonably possible. This will mean that at times the software will have to check to see whether the action to be taken is for an ADP system or not. There will also be a set of programs that are different from the non-ADP software due to the degree of change.

When multiple versions of programs are required, a second set of libraries (bind files, archives, headers, etc.) will be maintained. This will be done with totally consistent naming strategies applied throughout. Typical examples of such naming conventions would be inclusion of the strings "dps", "dpse", and "adp" in appropriate places in the names. Areas of the system where it is expected two versions of the software will be needed are:

clock reading
IOX interfacing software
interrupt handling and masking
fault handling
reconfiguration
console interfaces
BOS
page control

Areas of the system that will be expected to continue with a single version of the software are:

debuggers
compilers and assemblers
tools (dump analysis?)
error handling
segment control
resource management
network software
FNP software

Areas of the system that will require new software (and that will be identified with the "adp" sequence in names) will include:

interfacing to the SSF
interfacing to the hypervisor
interfacing to the performance monitor

Tracking of Bugs and Merging Systems

It is expected that all bug fixes applied to non-ADP systems will be applied in parallel to ADP versions of the software. A feature in the trouble reporting mechanism should be provided to facilitate this. It is also expected that new software developed at the time the ADP software is being developed should be done in such a way that it will work equally well on a non-ADP system as on an ADP system.

Logistics of Available Hardware

The software development schedules will be established so that, within a certain range of schedule error, the software will be done just as the hardware becomes available. There will not be a plan made to do the software and then sit back and wait for hardware. This places undo pressure on merging of bug fixes and tracking system changes in general.

It is expected that hardware will be available for initial bootload checkout in 3Q80 or soon thereafter.

In anticipation of problems in bringing up the hardware and software simultaneously, we are developing simulators for the processors, CIU, and IOX to allow us to debug a great deal of the ADP software before the hardware becomes available.

Conversion Scenarios and Plans

The only conversion programs currently proposed are needed to convert object segments that have old ITP format pointers. We do plan on providing utility programs that will point out potential problems as well as programs that will not execute at all. These latter cases are common in the system arena where we want to find all occurrences of privileged instructions, for example.

ANTICIPATED USE OF BOS

There is no plan to remove the BOS (Bootload Operating System) system from the standard Multics operational environment. The first set of mainframe software to be checked out on ADP will therefore be the BOS executive.

The various debugging utilities in BOS (DUMP, PATCH, ABS, TEST, etc.) will all be required for subsequent checkout of Multics on ADP.

There will be no reliance, initially, on any SSF software for the types of functions currently provided by BOS. Eventually, after better understanding of how best to use the SSF has been achieved, it is expected that some of these functions will be

moved from BOS to the SSF. However, the current plans call for all BOS functions to be supplied by BOS for first customer ship.

These plans, therefore, call for the SSF to do no more than simulate the current BOOTLOAD button as far as BOS and Multics are concerned. The SSF software will, of course, have to perform the various tasks which it alone can do, e.g. non-functional test, system maintenance panel, and system hardware initialization.

SPECIFIC TASKS AND ISSUES

This section describes the detailed work that must be done to bring Multics up under ADP. The order of tasks is not of significance in the following list.

New Calendar Clock Format

The RCCL instruction changes. This means that every use of it in the system must be found and one of two actions taken: either (1) convert the value to the old format before using the value, or (2) change relevant data bases to expect the data in the new format. Which of these is optimal will depend on the situation.

Nearly all references to the RCCL instruction in the user ring (user code) are via a PL/I builtin (clock) or via use of the clock_ subroutine. Changing these two parts of the system will get nearly all of the user-ring clock references.

A special library scanning tool will be made available for users (and used by system programmers) to help find potential problems with the RCCL instruction.

Address Trap Facility

A new and powerful feature made available with the ADP hardware is an address trap feature that allows users to gain control when a reference (of a particular kind) is made to a region of virtual memory. This capability will be enabled early in the ADP checkout cycle so that subsequent checkout can take advantage of it.

The initial mechanism will probably be enabled from BOS. Eventually, hopefully by first customer ship, the capability will be available to user-ring programmers through the standard debuggers.

I/O Changes Due to the IOX

There are several classes of changes required to the system that are due partially to the new IOX. These are:

1. Addition of more channels. Per-channel data bases will have to be made larger.

2. Changes in certain IDCW formats. Since the Multics "ioi_" software interface allows the use of IDCWs by user-ring programs, the redefinition of certain IDCW fields must be handled.

3. New interrupt levels. The entire strategy for partitioning interrupt priorities is new in the IOX. This will be reflected in much of the I/O initiating software where the interrupt "level" must be specified explicitly rather than implicitly as is the case with the IOM.

4. Linked mailboxes. Although there are currently no plans to take advantage of the mailbox linking facility of the IOX within Multics, all mailbox software must initialize these fields appropriately. (The specification of which mailbox to use at connect time will be used to facilitate dynamic reconfiguration of main memory.)

5. New connect protocol. All I/O initiating software will have to be changed to use the new IOX connect strategy. Similarly, all interrupt interceptors of I/O interrupts will have to be changed to interpret the interrupt and its associated data appropriately.

6. No CPI support. This is a problem due to the dependence of the ARPANet software relying on the ABSI channel based on the CPI interface. This is still an unsolved problem. (Currently about 35% of Multics sites are or will soon be on the ARPANet.)

These software changes will be incorporated in both BOS and Multics.

New Interrupt Mechanism

An entirely new interrupt mechanism is provided by the ADP system. The interrupt interceptor and the various interrupt handlers (to a lesser degree) will have to be changed to take the new strategy into account.

The major changes are: (1) different masking strategies, (2) introduction of the concept of eight interrupt levels with a queue of interrupts for each, and (3) the manner that interrupt "data" is acquired by the processor.

These software changes will be incorporated in both BOS and Multics.

Restrictions to Read-Alter-Rewrite

Only five instructions will retain the read-alter-rewrite behavior in the ADP system. These instructions are LDAC, LDQC, SZNC, STAC, and STACQ. Any software that relied on other instructions for indivisible memory modification (such as AOS or ERSA) must be changed. This affects user-ring software as well as supervisor code.

It is not anticipated that this will be a serious problem either in the supervisor or in user code. However, where it is a problem, it will be hard to detect as there is no automatic way a conversion tool can detect such use.

Dynamic Reconfiguration

The complete set of dynamic reconfiguration software will have to be redone. The main reason is that the hardware connectability and enabling of hardware data paths between major hardware modules is different in the ADP. In particular, they are eight instead of 32 interrupts, there is no SMIC instruction (CIOC must be used which can not be masked), CIUs are addressed explicitly for I/O, there is potentially more memory, and many registers that should be under control of the mainframe CPU are only changeable through the SSF interface.

All of these reasons will lead to a relatively large programming and design effort, not the least of which is specifying the operational interface for demand reconfiguration.

No More History Registers

The history registers, primarily an aid for checkout of hardware problems, are used in Multics user-ring environments as a debugging aid, primarily to track down "wild transfer" type software problems. Although this capability will be very well replaced with the transfer trace table provided in the ADP, several user-ring tools must be upgraded to understand and report on the new transfer table data. Much of this software is used by systems programmers -- some of it not officially installed.

New System Console

The ADP system will include a new console channel and the BOS and Multics systems must be changed to interface to it. The prime problem in this arena is specifying the operational interface that will be used. This is complicated by the existence of the SSF and the desire on the part of some customers to be able to run with no console at all.

The software protocol used when addressing the console will be the Common Exchange Protocol of DSA. Both BOS and Multics will be changed to use this protocol, as appropriate.

New BOOT Scenarios

The initial BOOT sequence will be different with ADP, being simulated by the SSF rather than the IOM. It is hoped that there will be minimal changes to the logistics of the BOOT, although currently the details are not known.

It is planned that the SSF will "boot" BOS and that BOS will "boot" Multics -- much as is done today.

Again, the operational interface -- including what consoles, etc. are used -- is the prime design problem at this stage. Currently, we are assuming the operator types BOOT on the SSF console and then never again looks at the SSF for normal operations.

New Cache Control

With the new fully hardware-controlled cache there will be no need for the software to worry about cache management. Basically, this means that certain pieces of software can be removed from the system. It is expected that this software change will be easy to do.

Redo All Relevant Tools

Due to the reformatting of so many internal registers many system tools will have to be upgraded. These tools range from online dump analysis tools to supervisor probing tools. There are scores of such tools and although each will not be hard to upgrade, the sheer numbers will lead to a significant programming effort.

As is the case with such tools, their value is highest at the initial debugging stage of a new supervisor checkout period and they should therefore be converted as soon as possible.

Enhance ALM, the Multics Assembler

Some relatively minor changes will be required of the assembler. At the least these will include understanding of all new opcodes, but changes should also include "adp" and "nonadp" modes (specified by control arguments or pseudo-ops) so that potential problems can be flagged at assembly time.

Hypervisor Software

The hypervisor software, not expected or needed for first customer ship, will have to be specified and written. This includes all interface specifications between the SSF as well as between BOS and Multics. In addition, all operational interfaces will have to be designed and specified. (It should be noted that the hardware support for the hypervisor can not be expected to be

correct and final until at least one software implementation is working on the hardware.)

Hypervisor interface within Multics

Once a hypervisor kernel is operational, Multics (and BOS) will have to be changed to interface to it. This means the Multics dispatcher will have to understand and run a "hyperswitcher" (a process that interfaces to the hypervisor). Accounting and metering changes will have to be made to both supervisor and user-ring code.

Again, specifying the design and interfaces is a required initial step before any programming work can be begun.

Interface to the SSF

The initial versions of BOS and Multics (first customer ship) will make minimal assumptions about the capabilities and availability of the SSF system. Initially, the SSF is needed only for booting the mainframe system -- transferring control to BOS. Additional SSF capabilities to which Multics (and BOS) must eventually interface include:

fault recovery
error logging
remote maitenance

It is expected that at initial BOOT time (when BOS gains control) the SSF will have given all configurable hardware over to the ADP mainframe system and that any reconfiguration that is done is completely under control of BOS and Multics. This means that no configuration data need be transferred between the SSF and the mainframe system and that all operational interfaces with respect to configuration are channeled through the mainframe system (Multics or BOS). The SSF software will be able to determine at any instance in time the exact configuration assumed and being used by Multics or BOS by examining the configuration data resident in main memory.

New Internal Register Formats

Several register formats change with the ADP. Other than user-ring tools (mentioned earlier) these changes will be of interest only to BOS and the Multics supervisor. Typical formats that change include:

DBRs
PTWs
SDWs
SCU data
I/O DCW and mailbox formats
fault register
mode register

history registers -> transfer trace table
interrupt mask registers
RSW data -> read reserved memory data

Redo of BOS

Most of BOS will have to be changed to some degree for ADP. The "utility" and "append" packages will be the heaviest hit, but the "setup" program (used during BOOT and communication between BOS and Multics) will also be heavily changed.

The DUMP, PATCH, and other debugging programs will be some of the first pieces of software necessary for ADP bootload checkout.

Generate Conversion Tools

The existence of conversion tools has been noted before as a requirement. The scanning program that looks for ITP pointers is the most obvious tool in this class, but other scanning tools will also be provided (to look for RCCL usage and any programs that include files with declarations of registers whose formats change).

The 6180 simulator, extended for ADP, will provide a useful tool for finding potential performance problems due to pipelined structure of the ADP. The simulator will also point out poor data/instruction layouts that would cause poor cache utilization in the ADP.

Interface to the Performance Monitor

The initial use of the performance monitor will not assume a direct hookup with Multics, rather, all data collection and reporting will be done externally to the mainframe system.

Paging Device Software

All paging device software (that currently interfaces to the bulk store) will be retained in the ADP system. It is not clear what, if any, new devices may fit into this kind of role, but bubbles and CCD devices should not be excluded.

DOCUMENTATION REQUIREMENTS

Many manuals will have to be upgraded or created as part of the ADP effort. For some manuals, as with the software, a single version will be acceptible for both ADP and non-ADP systems. Other manuals, however, will probably have to split into two versions. An interesting example of this last kind of manual is the "Hardware and Software Formats" PLM. We will generate an ADP-only version of this as part of the initial ADP system

development work (although it will probably not be published formally until first customer ship).

Other critically important manuals are those that operators need. The operational view of an ADP system will be different from earlier systems and it will be very important to describe not only what the operator interface is but also how the old interface maps into the new interface -- so current operators will have an easy transition to ADP.

ADP HARDWARE SIMULATOR

As part of the software checkout effort for ADP we are planning to develop an ADP system simulator that can take as input a virtual ADP system and sequence the system through state changes as a function of (1) the ADP instructions executed, and (2) external stimuli. This simulator will include the following major components:

- A CPU simulator consisting of a fault manager, appending unit, operations unit, etc.
- An IOX simultaor consisting of the IOX central logic, the channel logic for the various channels, etc.
- A CIU simulator consisting of the logic to manage the interrupt queues, connect protocols, memory, etc.
- An SSF simulator providing the bootload and maintenance functions.

This is a very large programming project that is only possible because of the Multics software factory capabilities. It will be a valuable tool for current and future performance analyses as well.

SECOND PHASE DEVELOPMENTS

Several developments were not included in the work planned for first customer ship. This is due mainly to the desire to keep the project simple and as bounded as possible. A list of some of the more important developments that must be worked on after first customer ship follows:

- compiler optimizations
  Once a reasonably accurate understanding of the performance and effect of the pipeline on performance is really understood, the various code generators will be changed to emit code sequences that are as optimal as is possible.
- multiprocessor configurations
  Although there should be little or no software changes required for multiprocessor configurations, there probably will be some changes required. These may include avoidance

of certain features or "padding" with NOPs as is done on current systems. We should not attempt to debug the cache while everything else is in a high state of flux.

- hypervisor
  The code of and the system code to interface to the hypervisor must be written.

- full SSF capabilities
  Once a fairly stable system is attained we should start to take fuller advantage of the kinds of things the SSF was designed for. These include T&D, remote maintenance interfaces, a consistent system logging capability, fault restarting, and the like. We should not attempt to include these in the first version of Multics ADP software.

PHASES FOR IMPLEMENTATION

The following phases of implementation specify what is currently planned. There are no dates given due to the extreme uncertainty of the hardware availability. Suffice it to say that when hardware is needed (Phase VII) we will be ready for it. This can only occur if this date falls after 2Q80 as it most surely will. It should also be pointed out that Phase IV, the actual conversion of the Multics system to run under ADP, will be delayed as long as possible (given a realistic understanding of the hardware availability) so that a prolonged period of merging and catchup is not required.

The following phases are in approximate chronological order. However, clearly several may be shuffled around for various reasons.

PHASE I

- Specify conversion/coexistence strategy for multiple versions of the system software.
- Implement any unimplemented software required to support multiple versions of the system software.
- Document the use of maintenance tools including things such as "open_program", macro processors, and new library structures and conventions.

PHASE II

- Design, specify, and document usage of the 6180 system simulator.
- Implement the 6180 system simulator.
- Design, specify, and document usage of the ADP system simulator (extended from the 6180 system simulator).
- Implement the ADP system simulator.

PHASE III

- Upgrade the Hardware and Software Formats PLM to include ADP formats for all appropriate registers.

PHASE IV

- Upgrade BOS to work under ADP.
- Debug BOS under the ADP system simulator.

PHASE V

- Upgrade Multics to work under ADP. This is the major task and includes the entire supervisor conversion. Due to the nature of Multics initialization the first code converted will be that which bootloads the system. Once the system boots successfully, the various user-ring packages will be converted.

- Debug Multics under the ADP system simulator.

- Document problems encountered with hardware design (as simulated).

PHASE VI

- Upgrade user-ring tools to work with the ADP system.

- Write, specify, and document conversion and scanning aids to be used in getting around incompatibilities.

- Upgrade debuggers to work with ADP formats and conventions.

PHASE VII

- Begin checkout of system (bootload) software on real ADP hardware. This is the first phase that relies on the ADP hardware being available.

- Bring BOS up to operational mode on real hardware. This will be the first real shakeout of the Multics appending unit. It will only checkout segmentation. The paging hardware will not be used until Multics is checked out.

PHASE VIII

- Begin checkout of the Multics supervisor on the real ADP hardware.

- Bring Multics up to command level on the real hardware.

- Debug BOS tools that need a Multics environment image under real hardware (much of this will have been debugged under simulation).

PHASE IX

- Design and implement changes to the reconfiguration software to work under ADP.

- Merge all changes to the system made to non-ADP releases.

PHASE X

- Measure performance of ADP instruction sequences.
- Change PL/I and FORTRAN code generators to use optimal code sequences.
- Change various "operator" segments to use optimal code sequences.

PHASE XI

- Run System M (or some such) as a Multics in-house service.

FUTURE PHASES

Other incomplete work includes integration of the hypervisor system, FTS, SSF and the like. These schedules are so far in the future at this time that it is probably counter productive to waste time trying to work out details.

TENTATIVE SCHEDULE

The following schedule of phases for the development necessary to get Multics running on the ADP hardware makes two assumptions. The first assumption is that the current ADP hardware schedule will be met. If the hardware is not available when planned, subsequent phases in this plan will have to slip accordingly. The second assumption is that the current plan which calls for Multics integrated hardware/software checkout not to begin until 2Q81 (given the current schedule) is ridiculous. Multics software should (must) be checked out on ADP as soon as possible and in parallel with the other operating systems.

Another point to make with respect to the Multics schedule is that Multics has not in the past gone throught System Environment Test (SET) and hence no one has gone through the trouble of setting everything up to do this. Probably this phase should be skipped and the Multics hardware made available for users at an exposure site as soon as the software conversion phase is complete.

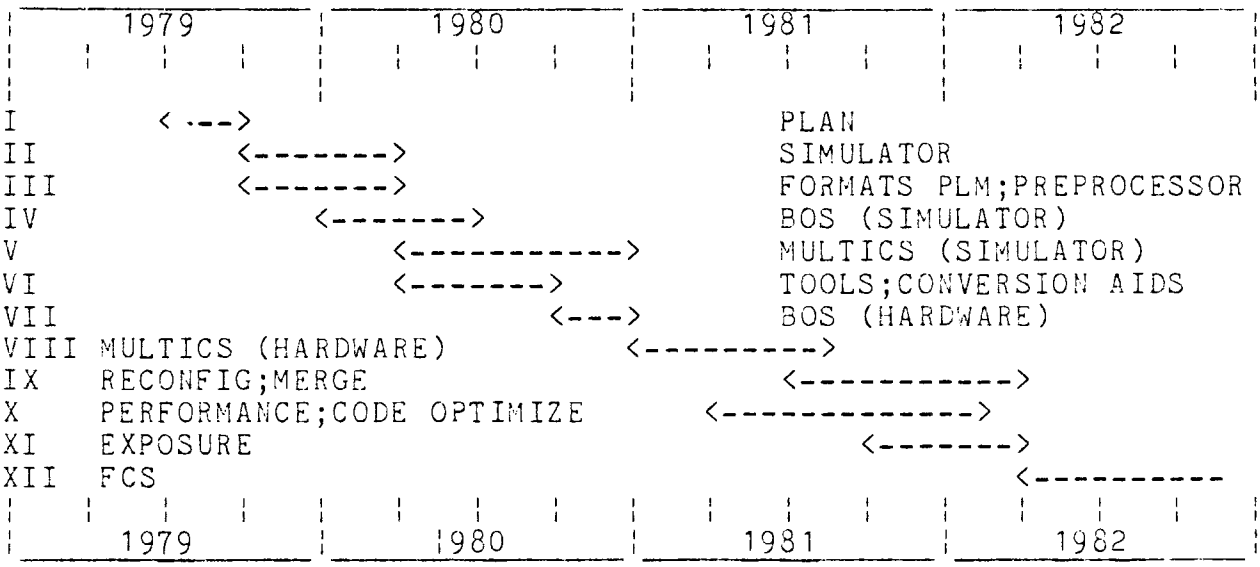

Table 2.
Software Development Schedule