ARPA ORDER NO. 2223

ISI/RR-77-58

March 1977

Benjamin Britt
Alvin Cooperband
Louis Gallenson
Joel Goldberg

PRIM SYSTEM: Overview

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

4676 Admiralty Way/ Marina del Rey/ California 90291
(213) 822-1511

THIS RESEARCH IS SUPPORTED BY THE ADVANCED RESEARCH PROJECTS AGENCY UNDER CONTRACT NO. DAHC15 72 C 0308. ARPA ORDER NO. 2223. PROGRAM CODE NO. 3D30 AND 3P10.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ISI/RR-77-58

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): PRIM System: Overview

5. TYPE OF REPORT & PERIOD COVERED: Research

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): Benjamin Britt, Alvin Cooperband, Louis Gallenson, Joel Goldberg

8. CONTRACT OR GRANT NUMBER(s): DAHC 1522 C 0308

9. PERFORMING ORGANIZATION NAME AND ADDRESS: USC Information Sciences Institute, 4676 Admiralty Way, Marina Del Rey, CA 90291

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order #2223

11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency, 1400 Wilson Blvd., Arlington, VA 22209

12. REPORT DATE: March 1977

13. NUMBER OF PAGES: 14

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): -----

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report): Approved for public release and sale; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES: This document supersedes and obsoletes PRIM Overview, ISI/RR-74-19.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number): ARPANET, emulators, emulation facility, microprocessor, microprogramming, microprogramming language, MLP-900, resource sharing, TENEX, time sharing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(over)

DD FORM 1473, 1 JAN 73 — EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT

This document is an introduction to the services available with the Programming Research Instrument (PRIM), an interactive microprogrammable environment available to remote users through the ARPANET. PRIM, which runs under the TENEX timesharing system, makes it possible to create and use emulators of existing or newly specified computers, with major emphasis on debugging tools. PRIM and TENEX together provide not only editors, compilers, and debuggers for creating emulators but also an environment for using the target systems, debuggers and configurors in the familiar language of the original system.

ARPA ORDER NO. 2223

ISI/RR-77-58

March 1977

Benjamin Britt

Alvin Cooperband

Louis Gallenson

Joel Goldberg

PRIM SYSTEM: Overview

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

4676 Admiralty Way/ Marina del Rey/ California 90291

(213) 822-1511

THIS RESEARCH IS SUPPORTED BY THE ADVANCED RESEARCH PROJECTS AGENCY UNDER CONTRACT NO. DAHC15 72 C 0308, ARPA ORDER NO. 2223, PROGRAM CODE NO. 3D30 AND 3P10.

ABSTRACT

This document is an introduction to the services available with the Programming Research Instrument (PRIM), an interactive microprogrammable environment available to remote users through the ARPANET. PRIM, which runs under the TENEX timesharing system, makes it possible to create and use emulators of existing or newly specified computers, with major emphasis on debugging tools. PRIM and TENEX together provide not only editors, compilers, and debuggers for creating emulators but also an environment for using the target systems, debuggers and configurors in the familiar language of the original system.

The present document supersedes *PRIM Overview*, ISI/RR-74-19.

PRIM SYSTEM: OVERVIEW

INTRODUCTION

The PRIM system is an interactive microprogrammable environment, available as a service facility to remote users through the ARPANET, that provides an integrated set of programming aids to create, debug, and execute programs for experimental computer environments. The TENEX timesharing system provides convenient access to editors and compilers for creating emulators, and PRIM also provides an environment for configuring and debugging target systems that can be operated by the user in the familiar language of the original target system. In general, the emulated machine accessible through PRIM provides better user debugging facilities and greater flexibility in system configuration than the actual machine, while still producing bit-compatible results on all levels of execution. PRIM is an attempt to generalize a solution to the problem of software development by means of emulation tools; it is a unique and powerful facility for improving software development.

PRIM is multi-access and simultaneous multi-emulator, with appropriate protection to prevent the errant user from interfering with other users. We currently are providing service for three emulated (target) computer systems: a UYK-20*, U1050**, and the Intel 8080 (chip). User manuals for these target systems, as well as for creating new emulations through PRIM, will soon be available for interested programmers and researchers. PRIM and the target systems will soon be available as tools under NSW (National Software Works).

THE PRIM FACILITY

The PRIM system, as a subsystem of TENEX [1,2], consists of an MLP-900 microprogrammable processor and appropriate software (PRIM Exec, PRIM debugger, GPM compiler, and TENEX MLP-900 driver) to create and interact with emulators or emulated systems. The PRIM Exec and debugger are table-driven so that the vocabulary and commands can be easily tailored to the familiar language of a specific machine or research application. The TENEX MLP-900 driver (in conjunction with the MLP-900 resident supervisor, or "microvisor") controls and provides access to TENEX for the emulators.

---

* Used by SDL (System Design Laboratory) of Naval Ocean System Center in San Diego.

** Used by LGS at Gunter Air Force Station.

Hardware

PRIM's hardware system, which has been reliably operating since 1974, is based on two processors: the shared use of a Digital Equipment Corporation PDP-10 with a TENEX operating system and the Standard Computer Corporation MLP-900 [3] prototype microprogrammable processor. The PDP-10 and MLP-900 share a paged memory containing 512K of 36-bit words as dual processors; the MLP-900 is also a device on the PDP-10 I/O bus. The MLP-900 has no peripherals of its own.

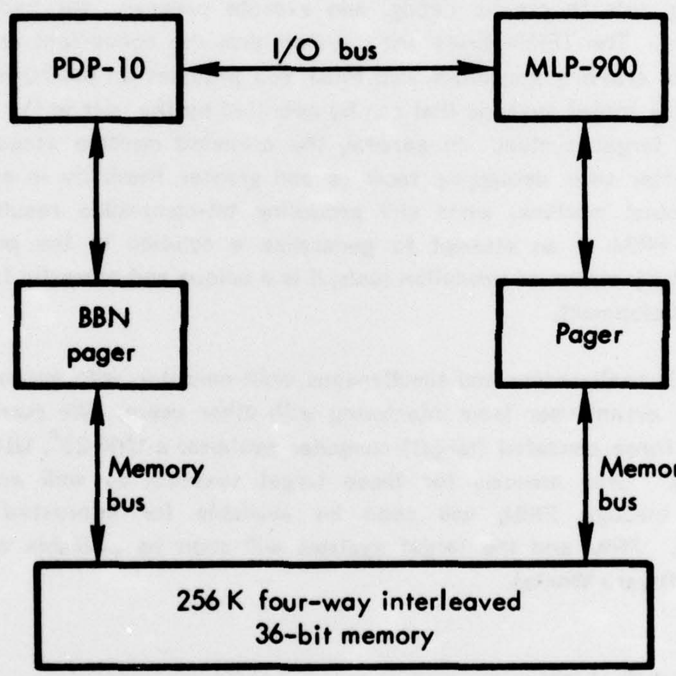

Figure 1.1 Basic PRIM configuration

The MLP-900 is a vertical word microprogrammed processor that runs synchronously with a 4-MHZ clock. It is characterized by two parallel computing engines: the Operating Engine (OE), which performs arithmetic operations, and the Control Engine (CE), which performs control operations (see Figure 1.2). The OE contains 32 36-bit general-purpose registers for operands and 16 36-bit mask registers to specify operand fields. A 1K 36-bit high-speed auxiliary memory is associated with the OE. The CE contains 128 state flip-flops, a 16-word hardware subroutine return stack, and 8 8-bit pointer registers. Additional hardware facilities of the MLP-900 are reserved for the microvisor state.

OPERATING ENGINE (I/O, arithmetic, logic)	CONTROL ENGINE (Branches, testing)
General registers 32 x 36 bits, R.0 - R.37	Flip/flops 128 x 1 bits, F.0 - F.377
Auxiliary memory 1K x 36 bits, A.0 - A.1777	Pointer registers 8 x 8 bits, P.0 - P.07
Mask registers 16 x 36 bits, M.0 - M.17	Subroutine stack 16 x 16 bits, S.0 - S.17
CONTROL MEMORY 4K x 36 bits	

Figure 1.2 MLP-900 configuration

The MLP-900 is accessible only through the PDP-10 as outlined above (i.e., the I/O bus and shared memory); it has no provision for direct connection of peripheral devices.

Software

The principal items of PRIM software are the General Purpose Microprogramming Language (GPM) compiler, the MLP-900 microvisor, TENEX MLP-900 driver, PRIM Exec, and PRIM debugger.

GPM and the GPM Compiler. GPM is a high-level machine-oriented language, designed explicitly for writing programs for the MLP-900. It offers a block structure and statement syntax similar to ALGOL. The compiler is capable of producing multiple instructions per statement.

MLP-900 Microvisor. The MLP-900 microprogram supervisor (microvisor) is a small, fully protected, resident system that controls the MLP-900 and its communication with the PDP-10. It loads and unloads the user's MLP-900 context upon command from the PDP-10, supports paging of the user target virtual memory, protects shared memory and the rest of the PDP-10 system from emulator errors, and mediates emulator calls on PDP-10 services. The microvisor interacts directly only with the user microcode and the TENEX MLP driver.

The TENEX MLP-900 Driver. Access to the MLP-900 from a TENEX process is done through the MLP driver in the TENEX Monitor. The driver is responsible for initializing the MLP-900 microvisor, controlling, scheduling and swapping users, accumulating accounting data, and passing along I/O requests. The microvisor is an extension of the driver; all communication with the MLP-900 goes through the driver, while communication with the driver occurs through the normal TENEX system calls.

PRIM Exec. The PRIM Exec provides the environment in TENEX needed to support each of the PRIM MLP-900 emulators and to allow the users of a particular (emulated) computer to access that environment.

The emulator support consists of the module responsible for controlling (emulated) execution, plus a module responsible for servicing the emulator's I/O requests. This I/O server offers the emulator access to logical target peripherals (files in the TENEX file system) or to the physical peripherals available through TENEX (such as local and remote terminals and magnetic tape drives).

A command language interpreter in the PRIM Exec (see Appendix) provides a uniform user interface modelled after the TENEX Exec, but with commands oriented toward the programmer familiar primarily with the computer being emulated. (The tailoring of the PRIM Exec, and also the PRIM debugger, to the details of a particular emulated machine, including its terminology, is accomplished through a set of machine-specific tables that accompanies each emulator.) Additional facilities are provided the emulator-writer for the development and checkout of a new emulator for the PRIM system.

The majority of the commands concern the building (and interrogating) of the emulated machine's configuration, e.g., installing devices on the machine, mounting TENEX files on those devices, and specifying miscellaneous parameters. In addition, there are commands that allow a complete checkpoint and subsequent restoration of the state of the emulation, the saving and restoring of various parts of the emulation context (e.g, target memory, or symbol tables,) and the generation of a transcript of all or part of a PRIM session.

PRIM Debugger. The PRIM debugger is a table-driven, interactive, symbolic debugger that permits a user of the PRIM system to debug target-machine programs in terms of symbols defined for the target machine, using the data representation and instruction formats of that machine. The debugger also provides symbolic access to the MLP context for both display and modification, with data representation under control of mode settings. Breakpoints can be set on location references or on events (predefined conditions) to interrupt the emulation; break-time debugger "programs" can be associated with breakpoints, to be automatically executed when the break condition is met. The debugger uses a command language with feedback and help available if needed. (See Appendix.)

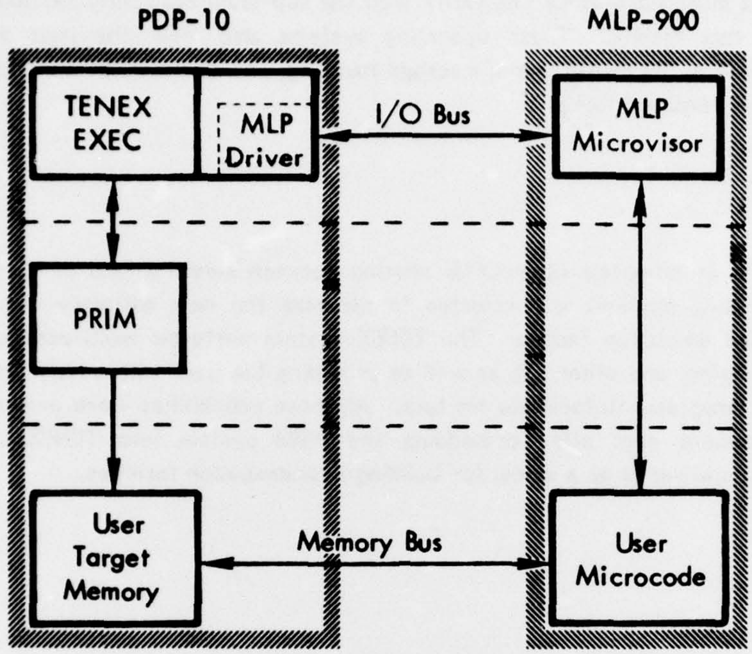

Figure 1.3 Basic PRIM software architecture

User Interface to PRIM

PRIM was designed for use by the programmer as well as the researcher interested in creating his own environment. To the programmer the system appears as the requested target machine (U1050, UYK-20, 8080 etc.,), the emulator being invisible. The programmer appears to have a stand-alone target machine and to be operating it from his terminal. The researcher creating his own microprogrammed environment, however, must be concerned with all three levels affecting operation within the PRIM system.

First, the researcher must know and understand the GPM compiler and the MLP-900 to generate microcode. Second, he must understand how the interface between the microvisor and MLP-900 driver is used to control the emulator and perform I/O. Third, he must understand the protocols that are followed by the PRIM Exec (and its I/O devices) to take advantage of all the services available in the PDP-10 through the MLP Exec and Debugger. This detailed knowledge required by the user is available in the appropriate user manual [4,5,6].

All users must have some familiarity with the top-level operating system (TENEX or NSW) to use this facility. These operating systems also offer the user many other capabilities (editors, file management, message handling, and all ARPANET functions) to help him perform the required tasks.

CONCLUSION

The PRIM architecture (dual CPUs sharing common memory, one of them having a complete operating system) was selected to minimize the new software required for a general-purpose emulation facility. The TENEX system performs multi-access tasks, file management, paging, and other I/O, as well as providing the user with a rich and broad set of well-tested programs to facilitate his task. All these capabilities were available at little or no development cost after embedding the PRIM system into TENEX. The PRIM architecture should serve as a model for building new emulation facilities.

APPENDIX

PRIM Exec Commands

CHANGE	Change keyboard interrupt characters
CLOSE	Close transcript file
COMMANDS	Runfile for commanding
DEBUG	Enter debug mode
ENABLE	Enable special features
FILESTATUS	List mounted files and status
GO	Start (continue) emulator from designated address
INSTALL	Install an I/O device
MOUNT	Mount TENEX file on I/O device
PERIPHERALS	Status of peripherals (installed device)
QUIT	Exit from PRIM
RESTORE	A saved environment (subset)
REWIND	A mounted file
SAVE	Saves environment (or subset)
SET	Parameters in registers, memory, etc.
SHOW	Parameters in registers, memory, etc.
SYMBOLS	Load symbol table

TIME	Display realtime, CPU times
TRANSCRIPT	Initiate a file for transcript of session
UNINSTALL	Remove I/O device
UNMOUNT	Remove file from I/O device

Debug Commands

All commands are recognized by their first character. In the syntax of the commands that follows, (...) is a complete syntactic unit and {...} is an optional input.

B	Break (event-list) or ((address-range)-list) <ESC> {(break-type-list)} {<ESC> (breaktime-commands)}
C	Clear-indicator-or-location ((address-range)-list)
D	Debreak-(clear breakpoint) (((address-range)-list) or <ESC>-all)
E	Evaluate (expression)
F	Format-specification (field-width-list)
G	Goto ((address) or not <uses program-counter>)
<LF>	Single-step
I	If (expression) <ESC>-then (command)
J	Jump-history-display {(length)}
K	Kill-symbols (symbol-list)
L	Locate {{NON } (value)} <ESC> {(mask)} <ESC> ((address-range)-list) {<ESC> (replacement)}
M	Mode-set ((mode-list) or <?>-check
N	New-symbols (symbol-list)
O	Open-program-symbols {(program-name)}

Q Quit-the-debugger (return to Exec)

S Set-indicator-or-location ((address-range)-list) {<ESC> (value)}

T Type-contents ((address-range)-list) {<ESC> (replacement)}

; Comment (one-line-of-text)

↑ Display-contents-of-prior-location

: Display-contents-of-same-location-in-current-output-mode

= Display-contents-of-same-location-as-bits-or-number

\ Display-contents-of-next-location

Further Commands Available to MLP-900 Programmers

- MLP-single-step

* Compile-MLP-microcode (octal-address) (GPM-statements)

/ Decompile-MLP-microcode

↑B (STX) Break-MLP-execution (octal-address)

REFERENCES

1. Bobrow, D. G., J. D. Burch, D. L. Murphy, and R. L. Tomlinson, "TENEX, A Paged Time-Sharing System for the PDP-10," *Communications of the ACM*, Vol. 15, No. 3, March 1972, pp. 135-143.

2. Meyer, T. H., J. R. Barnaby, and W. W. Plummer, *TENEX Executive Language Manual for Users*, Bolt Beranek and Newman, Inc., Cambridge, Mass., April 1973.

3. *MLP-900 Multilingual Processor--Principles of Operation*, Standard Computer Corporation, Santa Ana, Calif., 1970.

4. *PRIM System: User's Manual* (in progress)

5. *UYK-20 User's Guide* (in progress).

6. *U1050 User's Guide* (in progress).