Network Debugging Protocol — Eric Mader
RFC: 643 — July 1974
NIC #30873

This document describes a proposed protocol to be used in an implementation of a PDP-11 network Bootstrap service and a cross-network debugger. The protocol is designed for debugging processes running under an operating system which can perform some of the "lower level" debugging tasks, such as planting and removing breakpoints and single stepping. A subset of the protocol could be used if such a capability does not exist (a stand-alone program for example).

The protocol is a level 2 protocol, which bypasses the ARPANET HOST-HOST protocol. (This is implemented on TENEX using special privileged system calls which allow messages to be sent directly to and received directly from the IMP). Messages are sent between the PDP-11 and the remote debugger on one link (currently 377 octal). Each message from the remote debugger to the PDP-ll is a request that the PDP-11 perform some action, and each message from the PDP-11 to the remote debugger is either a reply to that request or an indication that a process has stopped running (i.e. has trapped, hit a breakpoint, etc). The exact format of the messages is shown in Figure 1 below.

Each command consists of an 8-bit op-code, and an 8-bit process-id, two 16-bit arguments, and an optional string of 8-bit bytes. The op-code field from the PDP-11 should be the same as that sent by the remote host to indicate successful completion of the request or be the same as that set by the remote host with the 200 bit set to indicate failure to complete the request. Op-codes from the PDP-11 which have the 100 bit set are asynchronous indication that a process has stopped for a reason other than a request from the remote host. (See description of asynchronous replies below). An op-code from the PDP-11 with both the 100 and 200 bit set is meaningless. Thus, the 8-bit op-code field sent by the PDP-11 can be thought of as a CAN'T Flag, an Asynchronous STOP Flag, and a 6-bit op-code.

In the description that follows the commands will be given as

NAME (Process-ID, Argument 1, Argument 2, BYTE STRING)
with only as many of the fields present as are used.

Op-Code 0 - NOP_____ _ _ ___

This command is intended to be used to determine if the PDP-11 is operational. It has no effect on any process running in the PDP-11. The response is NOP.

Op-Code 1 - DEBUG (Process)_____ _ _ _____ _____

This command requests the ability to debug the given process. The PDP-11 should respond with

DEBUGGING (process)

(op-code 1) if no other remote host is currently doing so, and

CAN'T DEBUG (process)

(op-code 201) if another remote host has been given permission to debug the process, or the process doesn't exist.

Op-Code 2 - END DEBUG (Process)

This command relinquishes the ability to debug a process. The PDP-11 should reply

END DEBUG (Process)

(op-code 2) unless the remote host isn't debugging the process, in which case it should respond

CAN'T END DEBUG (Process)

(op-code 202). If the process isn't running when the END DEBUG is done, then the effect should be the same as

RESUME (PROCESS)
END DEBUG (PROCESS)

Op-Code 3 - STOP (Process)

This command requests the PDP-11 to stop the given process from running. The PDP-11 should respond

STOPPED (process)

(op-code 3) unless the process isn't being debugged by the remote host. In this case it should reply

CAN'T STOP (process)

(op-code 203).

Op-Code 4 - DEPOSIT (Process, Address, Count, Byte String)

This instructs the PDP-11 to deposit the Count bytes in Byte String into the given process's address space beginning at Address. The PDP-11 should reply

DEPOSITED (Process, Address, Count)

if successful or

CAN'T DEPOSIT (Process, Address, Count)

if the deposit doesn't succeed, or the remote host isn't debugging the process.

Op-Code 5 - RESUME (Process)

This instructs the PDP-11 to allow the given process to resume running. The PDP-11 should reply

RESUMED (Process)

before it starts the process (in case it traps right away), or

CAN'T RESUME (Process)

if the process wasn't stopped, or the remote host isn't debugging the process.

Op-Code 6 - EXAMINE (Process, Address, Count)

This instructs the PDP-11 to return Count bytes from the given process address space starting at Address. The PDP-11 should reply

CONTENTS (Process, Address, Count, Byte String)

if it can supply the bytes, or

CAN'T EXAMINE (Process, Address, Count)

if all the bytes don't exist in the address space, or the remote host isn't debugging the process.

Op-Code 7 - DEPOSIT STATE VECTOR (Process, Index, Byte Count, Byte String)

This instructs the PDP-11 to deposit the Byte Count bytes in Byte String into the state vector for the given process starting with byte Index. See Figure 2 for the format of the state vector. The PDP-11 should reply

DEPOSITED STATE VECTOR (Process, Index, Byte Count)

unless the bytes won't all fit in the state vector or the remote host isn't debugging the process. In this case the reply should be

CAN'T DEPOSIT STATE VECTOR (Process, Index, Byte Count)

Op-Code 10 - BREAK (Process, Address, Proceed Count)

This instructs the PDP-11 to place a breakpoint at Address in the given process's address space, with a proceed count of Proceed Count. The proceed count can be changed by issuing another BREAK with a different count. The PDP-11 should reply

BROKE (Process, Address, Proceed Count)

if it was able to plant the breakpoint and

CAN'T BREAK (Proceed, Address, Proceed Count)

if it was unable to plant the breakpoint because the breakpoint tables for the process are full, or because the remote host isn't debugging the process. Note that the PDP-11 must keep track of where the breakpoints are, the old contents of these locations etc.

Op-Code 11 - UNBREAK (Process, Address)

This instructs the PDP-11 to remove the breakpoint from Address in the process's address space and release all storage associated with it. The PDP-11 should reply

UNBROKE (Process, Address)

when the breakpoint is removed, or

CAN'T UNBREAK (Process, Address)

if the process doens't have a breakpoint at Address or the remote host isn't debugging the process.

Op-Code 12 - SINGLE STEP (Process)

This instructs the PDP-11 to let the given process execute one instruction, using the Trace Trap feature. The PDP-11 should reply

SINGLE STEPPING (Process)

before the process is started, or

CAN'T SINGLE STEP (Process)

if the process is already running or the remote host isn't debugging the process.

Op-Code 13 - PROCEED BPT (Process)

This instructs the PDP-11 to allow the process to proceed from a break point trap (BPT, see below). The PDP-11 should reply

PROCEEDING BPT (Process)

before allowing the process to proceed, or

CAN'T PROCEED BPT (Process)

if the process isn't stopped at a breakpoint or the remote host isn't debugging the process.

Op-Code 14 - CREATE PROCESS

This instructs the PDP-11 to create a process (with its own virtual address space if the system will support such a feature). The PDP-11 should reply

CREATED PROCESS (Process)

if it could create a process, with the process-id in process. If the process creation failed or the system cannot create processes, the PDP-11 should reply

CAN'T CREATE PROCESS.

Once a remote host has created a process it is debugging it, that is a

DEBUG (process)

need not be done.

Op-Code 15 - DESTROY PROCESS (Process)

This instructs the PDP-11 to destroy the given process. After the process is destroyed the remote host cannot, of course, debug it so there is no need to do an

END DEBUG (Process).

The PDP-11 should reply

DESTROYED PROCESS (Process)

if the process has been destroyed. If the system doesn't support a multi-process structure or the remote host isn't debugging the process the PDP-11 should reply

CAN'T DESTROY PROCESS (Process).

Asynchronous Stop Replies

These replies (op-code between 100 and 177 octal) are sent by the PDP-11 wherever a process being debugged stops running for any reason other than in response to the

STOP (Process)

command. Each of these replies includes the entire state vector for the process.

Op-Code 100 - TRAP (Process, Reason, 0, STATE VECTOR)

This informs the remote host that a process has "trapped", or terminated abnormally. The reason is a system error code, or could be a trap vector address in the stand-alone case.

Op-Code 101 - HALT (Process, 0,0, STATE VECTOR)

This informs the remote host that the given process has terminated normally.

Op-Code 102 - BPT (Process, 0, 0, STATE VECTOR)

This informs the remote host that the given process has hit a breakpoint trap.

Op-Code 103 - TTRAP (Process, 0, 0, STATE VECTOR)

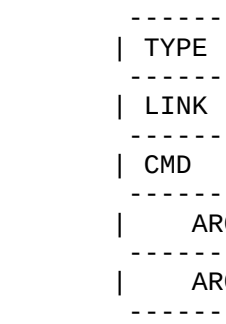
This informs the remote host that the given process has hit a trace trap, i.e. has just executed one instruction in response to the

SINGLE STEP (Process)

command.

Figure 1

FORMAT OF NETWORK DEBUGGER MESSAGES



TYPE -- The HOST-IMP/IMP-HOST message type, should always be zero.

HOST -- The source host ID.

LINK -- The debugger link.

STYPE -- The message subtype, always zero.

CMD -- The debugger command op-code, excess 100 for asynchronous reply’s, excess 200 for "can’t" reply’s.

PID -- Process ID of process being debugged.

ARG1 -- The first argument of the command.

ARG2 -- The second argument of the command.

Figure 2

FORMAT OF STATE VECTORS