

Subject: MCR-10037, Change system_startup_ to pass varying length arguments

Author: Gary Dixon

Date: January 10, 2018

Introduction

system_startup_ runs the Multics Ring 1 Initializer operator interface. It displays a Command: prompt, which allows the operator to enter an initializer command line which invokes one of 27 supported commands. This routine provides a very simple command processor which can process one command per line.

Problem

system_startup_ passes fixed-length arguments to the called programs (arguments declared as char(32) unaligned), rather than constructing an argument list describing the actual lengths of entered strings. In most cases, these fixed-length arguments are acceptable to the called subroutine, because PL/1 string comparison rules treat “ABC” and “ABC ” as being equal. However, some subroutines reject over-length arguments padded with trailing spaces.

For example, reload_volume rejects disk devices names that are over 8 characters in length, and is including the trailing spaces in arguments in that length calculation.

Proposed Changes

This problem will be corrected in system_startup_ (in >sl1>bound_system_startup_) by modernizing its code, as follows:

- As each argument is extracted from the operator-typed command line:
 - Place the argument in the existing storage, a 20-element arg array of char(32) items.
 - Record the actual length of the argument in a new argl array.
- Track the count of arguments as they are extracted.
 - Then the entire arg array need not be initialized to spaces.
- Convert the command variable, and the arrays of acceptable command names/abbrevs to varying length character strings, to make comparisons more efficient.
- In the call internal procedure:
 - overlay the storage for each argument with an overlay defined character string, referencing the corresponding argl element as the argument length.

- use the count of arguments (see above) to select one of 21 possible calling sequences (one for a subroutine with no arguments, the others for subroutines with from one to twenty arguments); then
- transfer to the correct calling sequence using a goto label array, based upon arg count.

Testing of the Change

These changes were tested, using the existing `system_startup_$read_command_entry` entrypoint. A test version of `reload_volume` subroutine was created, which display count, value, and length of each of its input arguments. The test `reload_volume` was invoked with 0, 1, 2, ..., 21 different arguments. Debugging data was checked to verify expected argument values and lengths were passed; the cases of 21 arguments caused the expected “Too many arguments given” error from `system_startup_`.

Bug Reference

- Reference URL of Multics Change Ticket: <http://multics-trac.swenson.org/ticket/96>

Documentation

Ring 1 user interface is not affected by this change, other than allowing valid commands to actually work as expected.

Version History

Date	Revision	Author	Comment
2018-01-10	1.0	Gary Dixon	Initial version of MCR.
2018-01-12	1.1	Gary Dixon	Change from using based character strings to using string overlay defining of the varying length character strings.