To: Distribution

From: Paul Green

Date: 01/04/77

Subject: Extending the Multics character set

This memo discusses how the Multics character set can be extended from the 128 characters of ASCII to the 512 characters supported by the hardware. It describes the current character set, the reason for extending the character set, and a method for extending the character set that causes no problems for existing software.

The Multics Character Set

Appendix A of the MPM Reference Guide describes the present Multics character set. It comprises the entire ASCII character set, with certain control characters reserved for future use. Since there are 128 ASCII characters, any character may be represented by a 7-bit code. When stored in a 9-bit byte, these characters are always right-aligned, and the first two bits must be zero. Any character with either of the first two bits on is refered to as a "non-ASCII" character.

Extending the Character Set

The Multics character set must be extended to include all possible 9-bit characters because Multics software (for example, APL, ARPANET, and EBCDIC-compatible programs) needs to use these characters. The MPM Reference Guide would not describe any particular extension of the character set, it would simply explain that all Multics software supports any 9-bit character. Unless the user is within a subsystem that "understands" the meaning of these additional characters, he or she would see the standard octal escape.

Proposed Changes

At present, nearly all PL/I constructs work equally well on ASCII and non-ASCII characters. String values containing non-ASCII characters may be defined in literals, assigned, concatenated, compared, and indexed. However, the search, translate, and verify builtins restrict their operation to ASCII characters. When non-ASCII characters are supplied as arguments, the result is not correct. Further, the collate and high

builtins are defined in terms of the ASCII character set.

The search, translate, and verify builtins can be changed to work for both ASCII and non-ASCII characters in a compatible fashion. No correct PL/I program can be affected by changing these builtins to work for non-ASCII characters. Therefore, it is proposed that these builtins generate tables that contain 512 entries, instead of the present 128 entries.

The collate and high builtins cannot be changed compatibly. The "new" collate would be 4 times longer than the present collate, although the first 128 characters would be identical. The "new" high would have an octal value of "777"b3 instead of "177"b3. Therefore, it is proposed that the present collate and high builtins remain unchanged, and that two new builtins, collate9 and high9 be added to the Multics PL/I language.

These are the only changes necessary to the Multics PL/I language and implementation to support a 9-bit character set. The changes to the rest of Multics are covered in a companion MTB, "Influence of APL on the Multics Character Set".

Terminology

I propose retaining the term "ASCII" to refer to the characters whose value lies between "000"b3 and "177"b3. This is consistent with the use of the term in the rest of the industry. I propose naming the full, 9-bit character set the "Multics Extended Character Set".