

BD.6.01

Identification

System Definition

J.A. Seltzer

Purpose

This section describes the definition of, and naming conventions for, the
Maetis system's operating system.



Discussion

A certain amount of organization is ~~more~~ essential in an operating system, so that one may identify, for accounting and debugging purposes, "which" system ^{or was} is in use at any given time. In Multics, this organization is formalized in a three-level system-naming scheme, and a library registry.

By identifying the three system levels and versions in use, and also the ^{name and version} identification of any public (and registered private) libraries, one can establish precisely ~~which copy of every~~ ~~what copy of any given program~~ ~~will be used~~ ~~belonging to that~~ what "total operating system" he is or was dealing with. When a large system may be dynamically updated by any of a number of system programmers, this formal user- and registry ^{control} ~~problem~~ is a prerequisite for sanity.

We first discuss the three system levels. A later section will describe the library registry.

System levels.

~~Multics is organized into~~
The various procedures of Multics fall naturally into one of three three

systems: the hard core system, the administrative system,

and the user interface system. In addition, there are any number

~~of~~ command and subroutine libraries, described later under library registry.

The three ~~major~~ systems mentioned above will, of course, come in

many variations, both major and minor. Each of these the three systems

has, ^{at least a 2-part component name} ~~therefore, a 2-part name~~, (two concatenated character strings),
with the ^{major component} ~~first part~~ a generic name for the system, and the ^{minor component} ~~second part~~
differing only between systems if significantly different, changed only when a major change is made, character

a version ^{identifier} ~~number~~, changed whenever any ^{part} ~~part~~ of the system is replaced or updated.

The Hardware System.

The hardware system has the property that ^{a single copy} it is shared among all users of a single shared-equipment complex. (Note that independent, though electrically connected, equipments may simultaneously support another different copy ~~part~~ of the same or different hardware system.) The version of the hardware system in use is ~~partly~~ determined by the Multics System Tape which was used to ^{initialize} generate the system, and by the system operator who selected one of several ~~versions~~ systems appearing on the ~~same~~ Multics System Tape.

The name of the hardware system is ~~derived~~ ^{constructed} as follows: stored

PL/I
~~part~~

in a structure, declared as follows:

```

dcl 1  hardcore-system-name

2  installation-code  char(24)  /* Unique name of installation running
                                this Multics. */

2  Tape-label        char(24)  /* Label from Multics System Tape
                                used to generate system. */

2  major-name        char(24)  /* name of the hardware system. */

2  version-name      char(24)  /* name of the version in use. */

2  time-loaded       bit(72)   /* calendar time system was
                                loaded. */

```



Then a typical hardware system name, if printed in a reasonable format,

might be:

install: MITMAC
 tape: MST74
 name: Multics/I
 version: 196
 load time: 1400.3, 6/27/68

~~While another system
 simultaneously ~~is~~ running
 Another system, under test on a separately configured section of the~~

~~the ~~the~~ installation hardware, might have the ~~the~~ name:~~

~~install: MITMAC
 tape: ~~MST74~~ MST141
 name: ~~Multics~~ Debug
 version: 855
 load time: 1714.6, 7/14/68~~

If the same system was in operation at another installation,

the installation name, and tape labels, ^{and boot time} would be different, but the major

and minor system names would be the same. A single installation

might, ^{normally} have two series of ^{hardware} system names, one, to denote the publicly

available system and a second (e.g., Multics/debug) to denote the a test

system run on a separately configured ^{hardware} section of the hardware.



It is essential to the operation of the scheme that the two
names "name" and "version" uniquely identify the system in
use, and a pair of names once used ^{must} never be used again to
identify a different system.

