

Identification

The Account Data Segment

T. H. Van Vleck

Purpose

The Account Data Segment (ADS) is used by the accounting subsystem to record resource consumption attributed to an account. There is one ADS for each account, which is shared by all processes which may cause changes to the account either as a result of their own use of processors, core, and input-output; or as a result of their changing the length of a file with the account's identifier in its directory entry.

The ADS is, of course, a file-system file, which is found in the Account Directory (described in BO.4.01) with a name which is the account identifier.

Section BO.4.02 describes the format and contents of the ADS.

Contents

The ADS contains three types of information:

1. "External" accounting information, which ties the account into the hierarchical account structure, and defines the availability of funds for the account.
 - a) account balance
 - b) withdrawl limit
 - c) tree name of superior account pool file
 - d) tree name of billing information file
2. "Internal" metering and control information, used by the metering sections of the supervisor to control and record raw resource use.
 - a) CPU cycles used (BO.4.01)
 - b) limit on CPU use
 - c) word-seconds of core used (BO.4.02)
 - d) limit on core usage
 - e) general i-o words transmitted (BO.4.04)
 - f) limit on general i-o

g) secondary storage i-o words transmitted (BO.1.03)

h) limit on secondary storage i-o for each device known to the file system (BO.1.06)

i) residence in words

j) limit on residence

k) total word-seconds charged

l) limit on word-seconds

m) length of device tables and, for each device in the tables

n) device id (BO.1.05)

o) device clock

p) device meter

q) device flag

r) device rate

s) device limit

3. Miscellaneous

a) "transactor tables" for such items as proprietary program use, reservation charges, etc.

b) number of processes using ADS

c) interlock

d) AMT index

e) time of last update

Explanation of contents

1. Balance - accounting and allocation within Multics is done in terms of a single accounting unit (an "information-processing-credit", perhaps, or even a dollar - the choice is up to the installation) which can be spent on any of the system resources. An accounting unit may be worth some number X of processor cycles, or some number Y of word-seconds of core, etc. The values are fixed by the pricing routines involved by "update-accounting."
2. Limit on withdrawals - an account is tied to a superior pool file which may contain an unallocated balance which any account dependent upon it may withdraw accounting units from, up to a limit specified in the account's ADS.
3. Tree name of superior - names this pool file
4. Name of person to be billed - this field contains a pointer into the billing list (BO.4.01) which describes where and how the accounts usage reports and bills are to be sent.
5. meters - see individual BO.1 sections for a description of how the metering figures are computed and placed in the ADS.

6. limits - in some applications, it is desirable to place a limit stop on a process's use of a particular resource, in order to prevent its "running wild." Whenever a meter value exceeds the corresponding limit, "update-accounting" will call a special exception routine, which the user may supply.
7. number of active users of account - this number is used for salvaging in the event of system malfunction, and as a flag to the billing programs that the account data ^{maybe} ~~is~~ unstable.
8. AMT slot number - this number is a relative index in the AMT (BO.3.07) to the AMT entry associated with the ADS. If the number of users is zero, it is meaningless.
9. Interlock - since several processes may be attempting to modify the ADS at once, it is necessary to prevent conflicts by requiring the interlock to be set before the ADS can be modified.
10. time last updated - the calendar clock reading at the last call to "update-accounting," ~~used for calculation of the time delta on secondary storage residence.~~

PL/I Implementation

The following PL/I statement
defines the Account Data
Segment:

del 1 ads ctl(p),

2 account bit (36),

2 balance fixed bin (71),

2 max-balance fixed bin (71),

2 (sc, lc, bc) fixed bin (17),

2 superior char (p → ads, sc),

2 log char (p → ads, lc),

2 biller char (p → ads, bc),

2 last_time bit (72),

2 lock bit (72),

2 anti bit (18),

2 users fixed bin (17),

2 delta,

3 processor fixed bin (71),

3 core fixed bin (71),

3 ss (2),

4 trans fixed bin (71),

4 time bit (72),

4 wd_secs fixed bin (71),

4 use fixed bin (71),

3 gim_io fixed bin (71),

2 use like delta,

2 limits,

3 processor fixed bin (71),

3 core fixed bin (71),

3 dedicated_dev fixed bin (71),

3 ss_io fixed bin (71),

3 qim_io fixed bin (71),

3 ss(2) fixed bin (36),

2 nio dev fixed bin (17),

2 dedicated_dev (p → ads, nio dev),

3 deviceid char (31),

3 flag bit (1),

3 price fixed bin (35)

3 clock bit (72),

3 charge fixed bin (72);