

**CDC 3300 COMPUTER OPERATING SYSTEM:
FILE ENVIRONMENT HANDLING METHOD**

Ákos Radó

Our task is to enlighten the file environment handling method of the CDC 3300 computer operating system, the search and retrieval of user files in the user files' maintenance system and after the analysis optimize the work of the computer.

The efficiency of MASTER 4.1 operating system's time sharing and multi-programming depends on optimum random access of mass storage.

The MASTER system operates in an environment in which all files have an identical basic structure. MASTER provides the user with a broad range of functions for manipulating the file definitions.

Functions that manage file definitions include allocation and release of space, modification of labels, expansion of defined file size, and opening and closing of files.

The system files required for MASTER to handle user files are File Label Directory (FLD) and Identifier File (IDF).

The FLD contains a complete description of each file known to the system. Each file has one file label entry written in the FLD of minimum 53 words in size (depending on the number of the segments of the user file). Each file has one two-word entry in the IDF. The first word is a 24 bit hash value calculated with "EXCLUSIVE OR" from the first ten words of the FLD. It is the remainder resulting from dividing the 40 characters of file identification (owner, filename, edition) by the largest prime number which will fit into 24 bits (8.388.593). Word two is the block number of the label's FLD entry. The block size of this file is set by a parameter of install time. The number of blocks is determined in the following manner:

- a./ The IDF file consists of two parts, a number of blocks which comprise the main body of the file, and an additional number of blocks

which comprise an overflow section.

- b./ The number of blocks in the main body is always a prime number. To arrive at this number, the maximum file count is increased by 10 % and divided by the number of entries per block (e.g. a 64-word block size has 32 entries). The next highest number in the list of prime numbers is selected as the number of blocks for the main body.

- c./ The overflow section is calculated as 10 % of the number of blocks in the main body of the file. For example, with a maximum file count of 1000, and a block size of 64 words, the main body contains 59 blocks, and the overflow contains 6 blocks, making a total of 65 blocks.

An entry in the IDF is made by dividing the remainder mentioned above by the number of blocks in the main body of the file. The remainder from this division plus one yields an IDF block number. If there is room in this block, the entry is placed here. Otherwise, the entry is placed in the first empty slot in the overflow area.

To reference a label, the owner, file name, and edition are divided by 8.388.593. This remainder is in turn divided by the prime number of blocks in the main body of the IDF. If no match can be made with any entry in this block, the overflow area is searched until the desired entry is found.

At allocation time the information about the file to be written into the FLD gets into the highest available block of the file. If the FLD is full, we try to find an empty block - due to a previous deletion of a file - that is the FLD is not compressed after deletions.

In the IDF the searching algorithm depends on the fact whether we allocate a new file or we look for an existing file (opening, deletion, modification). In the latter case the algorithm is as follows: the overflow area is searched serially, if not found, the primary area is searched.

In the first case we begin to search an empty slot in the primary area, then it not found in the overflow area. The search in the primary area is as follows: in the block, the number of which counted from the hash code, serially (bucket!) look for an empty slot or a slot where previously an already deleted file resided. Deleting a file the first word of its entry in the IDF is zeroed, the second word (reference to the block in the FLD) remains unchanged (reason to be seen later).

In case of allocation it may happen, that we find however an empty slot in the primary or overflow area but the FLD is full (because it is not compressed). In this case we examine one by one the blocks of the IDF, whether there is an entry having its first word zeroed. If found, the information about the newly allocated file written into the FLD block, whose number was just found in the second word of the IDF entry. The IDF entry's second word is zeroed as well, and the two word entry is placed into the primary or the overflow area.

The primary hash function of the addressing algorithm is the division method. The secondary function has three phases: open addressing linear search in a bucket of the primary area; if it is not successful, then open addressing linear search serially in the blocks of the overflow area; if it is not successful, open addressing linear search serially in the other blocks of the overflow area.

By this relatively simple, but sometimes too long algorithm ensured, that the algorithm always finds an empty slot.

The MASTER 4.1 operating system has been installed to handle maximum 4291 files (block number of FLD). To enlarge the FLD needs new installation of the whole operating system (about 50 hours computer time).

To define the IDF size, the installation guide book suggests the following: at first we fix the block size (bucket size) which should be divisible by two and less than 160 words (one sector on magnetic disc - if greater, then data transfer between the disc and the memory (30 msec) at least would increase by two).

In the existing installation block size is 160 words, that is one block can accommodate 80 entries.

Block number is defined as follows: increase the maximum file count (4291) by 10 % (4720) then divide by the number of entries (80) and from the following prime numbers (3,7,11,31,59,127,503,1019,2039,4091,8191,16319,32719,65519) which is closest one but less than the result of the previous division (in our case: 59). The form of these prime numbers is

$$4j + 3 \quad (j \text{ integer})$$

which ensures a uniform distribution of the files over the available addresses.

These 59 blocks consist of the primary area. The overflow area is 10 % of the primary area (in our case 6).

According to the installation instructions the bucket size is between 2-160 words. Because the IDF and FLD reside on magnetic disc, our task is to minimize the data transfer between the memory and the magnetic disc (memory cycle time by four magnitudes less than data transfer time!).

To execute the search algorithm - excluding data transfer - we need about 1 msec (it varies with loading factor of the files).

The CDC 3300 computer operates in three shifts, 5 day a week (300-400 jobs per day) and about 50 allocation and deletion occurs daily. File data modifications are about 10-15 daily. File openings are about 1000 daily.

To analyze the quality of the system, the following data are of particular interest:

- (α) loading factor at the overflow area
- dislocation (displacement distance from the "originally" appointed block)
- the effect of insertion/deletion cycles on the previous data

Our system to be analyzed had 4182 files (loading factor 97,5 %). In

the course of the first measurement no deletions occurred.

If $\alpha < 90\%$ no entries were occupied in the overflow area. If $\alpha = 90\%$, one primary block became full, that is the probability that an overflow entry becomes occupied equals $\frac{1}{59} = 0.0169$.

This result well coincides with the theoretical approximation. The effect of insertion/deletion cycles was investigated in the $[0.1;0.9]$ interval of α . The 40 characters of new files were generated by random number generator. One note before we evaluate the results: in optimal case one access to a file needs two data transfers (60 msec). This time is about 0.1 % of the daily work of the computer. Table 1 shows the effect of insertion/deletion cycles at different loading factors on the percentage increase of data transfers.

Insertion/deletion cycles

α	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
0.7	0.8	0.9	1.9	2.8	3.3	3.9	4.1	4.3	4.4	4.6
0.8	2.1	2.3	4.7	6.7	8.0	9.4	9.9	10.2	10.2	11.0
0.9	1.5	2.9	5.9	8.2	9.9	11.7	12.3	12.6	13.2	14.6
0.975	23.8	35.7	43.0	71.4	78.5	85.7	87.0	88.1	94.0	100.0

In the $\alpha [0.7;0.9]$ interval no essential increase. If $\alpha > 0.9$, and cycle number is in the magnitude or larger than the table size the data transfers are doubled. No full overflow area and for this reason dislocation and overflow loading factor could be easily deducted from data in Table 1.

The effective operation of the almost full IDF and FLD could be assured by the occasional run of *SF4 operating system program which insures the compression of the FLD and IDF.

According the above results, if the increase of data transfers exceeds 70 %, we have to run our program (at present about once in 5 weeks).

However the compression of the two files apparently increases slightly the system's throughput, but taking into account, that job execution is suspended while data transfers executed, and the two file is single accessible, the effect of the compression is larger.

Ö s s z e f o g l a l ó

A CDC 3300 számítógép operációs rendszere file kezelő módszere

Radó Ákos

Analizáljuk a MASTER 4.1. operációs rendszer file kezelő módszerét. Vizsgáljuk a file behelyezés/kivétel ciklusok hatását a túlcordulási terület kitöltöttségére és a file bejegyzések diszlokációjára. A rendszer hatékony Működéséhez szükséges file bejegyzések átrendezésének gyakoriságát a mérési eredmények alapján adjuk meg.

Резюме

Метод обработки файловой системы операционной системы ЭВМ CDC 3300

Акош Радо

Анализируется метод обработки файловой системы операционной системы MASTER 4.1 ЭВМ CDC 3300. Качество системы кодирования hash измеряется оценкой влияния циклонов ввода/вывода на сдвиг и на коэффициент нагрузки области переполнения. Для оптимальной работы ЭВМ диапазон времени между двумя разборками файлов системы определяется.