A model of the process of writing and deleting file information on a disk with NTFS

A. CHOJNACKI, F. DARNOWSKI
andrzej.chojnacki@wat.edu.pl, darnowski.fryderyk@gmail.com

Military University of Technology, Faculty of Cybernetics
Institute of Computer and Information Systems
Kaliskiego Str. 2, 00-908 Warsaw, Poland

This paper aims at demonstrating a mathematical model of the process of writing and deleting information about files on a disk, using the contents of the $MFT system file, i.e. in a file generated in the NTFS (New Technology File System). The presented model uses the language of control theory, where the state of the system is equal to the state of the disk and the state of the $MFT file, and where control is understood as undertaking the action of writing or deleting. The deterministic nature of the process and its stationarity were assumed. Then, based on the transition function after its specification, we suggest constructing further inverse images of possible prior states at subsequent stages of data writing or deletion. The obtained results form the basis for the implementations developed.

Keywords: hard drive, NTFS, $MFT.

DOI: 10.5604/01.3001.0013.6602

1. Introduction

There are many methods of data retrieval from storage devices [1]. This also accounts for disks in which the files are organized in accordance with the widely used Microsoft Corporation standard for Windows operating systems starting from NT 3.1 that is with the NTFS (New Technology File System), replacing the FAT file system (File Allocation Table) used previously with smaller disks. What is characteristic for the NTFS is that it uses system files, and among them, the $MFT system file (Master File Table) as the most important system file from the point of view of data file distribution. The information found in that core file, hereinafter referred to as the $MFT table, indicates the state of file attributes on the disk at the moment of analysis, but it may also contain information regarding data stream writing and deletion. Such an example process is described in [2], with a simultaneous analysis of the content of the $MFT table and disk memory. The conclusions drawn in [2] have led us to draw a model allowing for the implementation of this process in order to examine its properties from the point of view of its usability in data retrieval. The aim of this model is therefore the mathematical description of the changes of information within the $MFT table and on the disk resulting from a sequence of writings and deletions of data files in such a way that we can analyze how this process might have proceeded, based on a set of "reverse" processes for a given $MFT table and drive content. The process of data writing and deletion is, under certain circumstances, absolutely unequivocal. This, however, cannot be said about the "reverse" process, and thus only potentially obtainable sets can be the result of the analysis. Nevertheless, in many cases, these are sufficient to locate the lost data on the disk, especially when the analysis is carried out by a person with additional information regarding the lost data.

2. Assumption

We will only consider files of written or deleted by user, hereinafter referred to as the files, contrary to the $MFT table, which, from the point of view of the operating system, is also a file. Let us consider a deterministic file-writing process in which each time we know the size of the file to be written and we have all the necessary information regarding the distribution of the data on the disk and the $MFT table. This means that we are not analyzing a case of stream writing, in which the size of the file written is unknown, and the location is assigned dynamically. Such case is described in detail in [3]. Comprehensive knowledge of the

contents of the disk and the $MFT table also take in the case of data deletion. We have excluded from the analysis issues related to multitasking, data buffering, defragmentation and other features characteristic of disks with operating systems installed. That is why we omit those disk fragments used by the operating system or occupied by system files or folders.

Let us further assume that each file being written has a cluster fold size and all its attributes fall into one $MFT table record. When writing a file, its attributes are located in the first free record of the $MFT table, i.e. in a record where the deletion tag equals 1, and the record had not been used previously. In the latter case, another $MFT record will be created.

We assume a finite period of time in which writing and deletion operations are performed on the disk in a serial manner, and the physical parameters of the disk do not change over the given period of time. A subsequent operation can only start when the previous one has come to an end. We assume the discretization of time, where the numbers of subsequent moments are related to the writing or deletion operations undertaken. Therefore, those numbers correspond to the numbers of time periods in which subsequent operations are carried out.

The data file to be written is saved in one area, but this area can consist of many fragments located in various parts of the disk. File writing takes place in accordance with the BFA algorithm (Best Fit Algorithm). This means placing the file in one compact area of the size closest to the size of the file to be written. If there exists more than one such area, the one that is closest to the beginning of the disk is selected. If such a compact area does not exist, then the fragment of the file is written in the largest available compact area, and the rest of the file is written in accordance with the BFA principle above. This way, the file remains fragmented, and the information about its location is stored in one $MFT table record. It might also happen that the file to be written is larger than all the disk fragments available for writing. In such cases, we assume that the file is omitted, another writing and deletion operation is undertaken, and the system administrator is informed about the situation. Examples of the above process are presented in [2] and detailed explanation of the file system is presented in [4].

3. Description of the $MFT table

We divide the time period in question into $T+1$ ranges, whose length can vary. Let $\mathbb{T}=\{0,1,2,\ldots,T\}$ denotes the set of numbers of those time periods. The beginning of the time period numbered 0 denotes the period of time for which we wish to locate the lost data, and time period T denotes the moment when we start data retrieval. The beginnings of time periods from zero to $T-1$ denote undertaking the action of writing or deletion.

Let K denotes the maximum number of records that the $MFT table can contain. Below, we list the attributes within the k-th $MFT table record and analyzed at the beginning of the t time period:

- deletion counter, denoted by $L_t^k \in \mathbb{N}$ (where $\mathbb{N}=\{0,1,2,\ldots\}$ denotes a set of natural numbers); the value of the deletion counter denotes how many times the files, whose attributes have been allocated in the k-th record, have been deleted (from the beginning of the t time period, accounting for the time before the time period in question);
- deletion counter, whose value is denoted by $B_t^k \in \{0,1\}$, where $B_t^k=1$ denotes that k-th record has been deleted in the time period prior to the t time period or before the $t=0$ time period, and after its deletion it has not been overwritten, and $B_t^k=0$, denotes the reverse situation.

Of course, notations proposed above, are based on real physical data that can be read from $MFT file.

The value of the deletion counter must not decrease with time, which means that for the saved $MFT table record:

$$\forall t',t''=\overline{0,T}:\left(t'<t''\Rightarrow L_{t'}^k \le L_{t''}^k\right) \qquad (1)$$

An attempt to write a file results in data writing of this file in one or more fragments, or a failure to do so, if the size of the available disk areas is too small to fit the file.

Let $MD \in \mathbb{N}_+$ (where $\mathbb{N}_+$ denotes a set of positive numbers) denotes the size of the disk in clusters numbered with a variable, assuming values from 1 to MD, with each pair of adjacent clusters numbered with a pair of consecutive numbers.

Let $R_t^k \in \mathbb{N}_+$ denotes the number of fragments on which the file presented in the k-th record of the $MFT table is written. The value $R_t^k=1$ means that the file was written in one compact area. Otherwise, the file was fragmented into $r=\overline{1,R_t^k}$ numbered

variables. The r-th file fragment has been written in the part of the disk beginning with the cluster numbered according to $PI_t^{k,r} \in \mathbb{N}$. The written fragments may be of various length, measured in the number of clusters. Let us denote it with the symbol $WI_t^{k,r} \in \mathbb{N}_+$. In this case, we may assume that the area occupied by the file, whose attributes are in the k-th record of the \$MFT table, can be shown as:

$$I_t^k \triangleq \left\langle \left\langle PI_t^{k,1}, WI_t^{k,1} \right\rangle, \left\langle PI_t^{k,2}, WI_t^{k,2} \right\rangle, \ldots \right.$$
$$\left. \ldots, \left\langle PI_t^{k,R_t^k}, WI_t^{k,R_t^k} \right\rangle \right\rangle \tag{2}$$

Nevertheless, these are disconnected fragments located on the disk, thus, they meet the following condition:

$$\left(\forall r = \overline{2, R_t^k} \cdot PI_t^{k,r-1} + WI_t^{k,r-1} \le PI_t^{k,r} \right) \wedge$$
$$\wedge \left(PI_t^{k,R_t^k} + WI_t^{k,R_t^k} \le MD \right) \tag{3}$$

The state of the k-th record of the \$MFT table after the action performed in the t time period (labelled as M_t^k), can thus be shown as:

$$M_t^k = \begin{cases} \varnothing \text{ if } k\text{-th record of \$MFT table is not} \\ \quad \text{filled in the } t\text{-th time period}, \\ \left\langle L_t^k, I_t^k, B_t^k \right\rangle \text{ in reverse situation.} \end{cases} \tag{4}$$

File fragments whose attributes are in the \$MFT table, and which were not identified as deleted, that is all those file fragments whose names are in the filled \$MFT table records and have not been deleted, must not occupy the same disk clusters. This applies only to files whose descriptions are in the \$MFT table records, which are files with numbers from the following set:

$$\mathbb{E}_t \triangleq \left\{ k = \overline{1, K} : M_t^k \ne \varnothing \wedge B_t^k = 0 \right\} \tag{5}$$

Therefore, the following condition must be met:

$$\forall t = \overline{0, T} \forall k', k'' \in \mathbb{E}_t \forall r' = \overline{1, R_t^{k'}}, r'' = \overline{1, R_t^{k''}} :$$
$$\left((k' \ne k'' \vee r' \ne r'') \Rightarrow \right.$$
$$\Rightarrow [PI_t^{k',r'}, PI_t^{k',r'} + WI_t^{k',r'}) \cap \tag{6}$$
$$\left. \cap [PI_t^{k'',r''}, PI_t^{k'',r''} + WI_t^{k'',r''}) = \varnothing \right).$$

If an empty k-th \$MFT table record is filled due to an action undertaken in the t-th time period, then it will remain filled also in all subsequent time periods.

The name of the file, labeled as $A_t^k \in Z(\alpha)$, becomes an attribute of the file written in the k-th record in the t-th time period (where $Z(\alpha)$ is the set of allowable words created from the α, which can be the names of the analyzed files, including folder names, sub-folders and file types). As the k-th record may not be filled, the attribute of the file written in it, labeled as D_t^k, equals:

$$D_t^k = \begin{cases} \varnothing \text{ if } M_t^k = \varnothing, \\ A_t^k \text{ in reverse situation.} \end{cases} \tag{7}$$

The name of the file must not repeat in various \$MFT table records corresponding to undeleted files, hence the following condition is met:

$$\forall t = \overline{0, T} \forall k', k'' = \mathbb{E}_t : \left(k' \ne k'' \Rightarrow A_t^{k'} \ne A_t^{k''} \right) \tag{8}$$

The file described in the k-th \$MFT table record analyzed in the t-th time period was written in that time period or earlier. The number of the time period in which it was written, is labeled as tz_t. Obviously, $0 \le tz_t \le t$. In summary, it may be assumed that the k-th \$MFT table record is described in the t-th time period by:

$$\left\langle tz_t, M_t^k, D_t^k \right\rangle \tag{9}$$

whose elements are defined with formulas (2), (4) and (7), that meet criteria (1), (3), (5), (6) and (8).

4. Description of the disk

Those areas of the disk with the allocated files of file fragments which are not deleted at the beginning of the time period t, are the area whose fragments cannot be overwritten after undertaking the action of writing. The set of the numbers of those areas that meet criteria (3) and (6), will be labeled as $O_A^t \in \mathbb{N}_+$, that is:

$$\mathbb{O}_A(t) \triangleq \bigcup_{k \in \mathbb{E}_t} \bigcup_{r=1}^{R_t^k} [PI_t^{k,r}, PI_t^{k,r} + WI_t^{k,r}) \tag{10}$$

Some of the disk areas occupied by files with different names will be compact, whereas some of those areas will be separated by fragments unallocated any to of the deleted files.

Let us label the number of compact areas in the $\mathbb{O}_A(t)$ set as $O_A^t \in \mathbb{N}_+$; the beginning of the o-th of them as $PA_t^o \in \mathbb{N}$; and its length in clusters as $DA_t^o \in \mathbb{N}_+$. Having assumed that, we can depict the $\mathbb{O}_A(t)$ set as a sequence of

descriptions of the time periods:

$$\mathbb{O}_A(t) \triangleq \left\langle \left\langle PA_t^1, DA_t^1 \right\rangle, \left\langle PA_t^2, DA_t^2 \right\rangle, \ldots, \left\langle PA_t^{O_A^t}, DA_t^{O_A^t} \right\rangle \right\rangle \quad (11)$$

ordered based on their beginnings, that is:

$$\forall o = \overline{2, O_A^t} : PA_t^{o-1} < PA_t^o. \quad (12)$$

Based on (3) and (6) we can also infer that:

$$\forall o = \overline{2, O_A^t} : PA_t^{o-1} + DA_t^{o-1} \le PA_t^o \quad (13)$$

Entries (10) and (11) will be treated as equivocal.

The BFA algorithm, which following the initial assumption is responsible for the process of file writing, considers only those disk clusters that are not occupied by files described with records with numbers that do not belong to the $\mathbb{E}_t$ set, i.e. are located outside the $\mathbb{O}_A(t)$ set. The disk area accounted for by the BFA can be divided into the following areas:

- disk areas with files of file fragments that were deleted but not overwritten; this area will be referred to as $\mathbb{O}_D(t)$;
- disk areas that were not considered in the \$MFT record table in the time period t; these areas will be referred to as $\mathbb{O}_U(t)$;
- disk areas occupied by system files or occupied by the system – following the initial assumption, these areas are not taken into account, which is reflected in the way of numbering the clusters.

All of the above allows us to state that:

$$\mathbb{O}_D(t) \cup \mathbb{O}_U(t) = \{1, 2, \ldots, MD\} \setminus \mathbb{O}_A(t) \quad (14)$$

moreover,

$$\mathbb{O}_D(t) \triangleq \left\{ m = \overline{1, MD} : \exists k \notin \mathbb{E}_t : \left(B_k^t = 1 \wedge \exists r = \overline{1, R_t^k} : \left(PI_t^{k,r} \le m < PI_t^{k,r} + WI_t^{k,r} \right) \right\} \setminus \mathbb{O}_A(t) \quad (15)$$

as well as

$$\mathbb{O}_D(t) = \{1, 2, \ldots, MD\} \setminus \left(\mathbb{O}_A(t) \cup \mathbb{O}_U(t) \right). \quad (16)$$

Conditions (15) and (16) mean that the areas $\mathbb{O}_A(t)$, $\mathbb{O}_D(t)$ and $\mathbb{O}_U(t)$ are disjointed pairs, and what is more, their sum equals $\{1, 2, \ldots, MD\}$.

Similarly as in the $\mathbb{O}_A(t)$ area, let us organize the compact fragments belonging to the $\mathbb{O}_D(t)$ and $\mathbb{O}_U(t)$ areas in sequences as follows:

$$\begin{aligned} \mathbb{O}_D(t) &\triangleq \left\langle \left\langle PD_t^1, DD_t^1 \right\rangle, \left\langle PD_t^2, DD_t^2 \right\rangle, \ldots, \left\langle PD_t^{O_D^t}, DD_t^{O_D^t} \right\rangle \right\rangle, \\ \mathbb{O}_U(t) &\triangleq \left\langle \left\langle PU_t^1, DU_t^1 \right\rangle, \left\langle PU_t^2, DU_t^2 \right\rangle, \ldots, \left\langle PU_t^{O_U^t}, DU_t^{O_U^t} \right\rangle \right\rangle, \end{aligned} \quad (17)$$

where PD_t^o, PU_t^o denote the beginnings of fragmented areas respectively, DD_t^o, DU_t^o denote their lengths, and O_D^t, O_U^t the numbers of those disjointed fragments. If we present the above sets in this way, parallel conditions must be met for (12) and (13), that is:

$$\begin{aligned} \forall o = \overline{2, O_D^t} &: PD_t^{o-1} < PD_t^o \\ \forall o = \overline{2, O_U^t} &: PU_t^{o-1} < PU_t^o \end{aligned} \quad (18)$$

as well as

$$\begin{aligned} \forall o = \overline{2, O_D^t} &: PD_t^{o-1} + DD_t^{o-1} \le PD_t^o \\ \forall o = \overline{2, O_U^t} &: PU_t^{o-1} + DU_t^{o-1} \le PU_t^o \end{aligned} \quad (19)$$

The $\mathbb{O}_D(t)$ and $\mathbb{O}_U(t)$ sets describe the sets of cluster numbers that may be used by the BFA algorithm, hence we may assume that for the purpose of our analysis, the ordered triple:

$$\left\langle \mathbb{O}_A(t), \mathbb{O}_D(t), \mathbb{O}_U(t) \right\rangle \quad (20)$$

is a sufficient description of the state of disk memory in the t-th time period in which this state does not change. It should also be noted at this point that in the t-th time period, the above sets are clearly defined by the \$MFT table records, and the knowledge of any two of those sets allows to determine the third one.

The operations of the BFA algorithm on the $\mathbb{O}_D(t)$ and $\mathbb{O}_U(t)$ sets result from one more premise. Namely, in the case of file writing on the disk, what happens first is the physical writing of the data on the disk, and only then is the \$MFT table updated.

5. The description of the state of the process

Let S_t denotes the state of the \$MFT being filled and the disk after the operation of writing or deletion at the beginning of the $t = \overline{1, T}$ time period. Let S_0 denotes the beginning state of the \$MFT table and the disk which we would like to get to know, at least partially.

The state of S_t $(t \in \mathbb{T})$ will be described by attributes in each \$MFT table record, by the information regarding the location of the data on

the disk corresponding to those records and by submitting the cluster sets that the BFA algorithm will be able to operate on. In sum, the S_t state can be presented as:

$$S_t \triangleq \left\langle t, \left\langle M_t^1, D_t^1 \right\rangle, \ldots, \left\langle M_t^K, D_t^K \right\rangle, \mathbb{O}_A(t), \mathbb{O}_D(t), \mathbb{O}_U(t) \right\rangle \tag{21}$$

This way of presenting the state means that some pieces of information are repeated, which results from the conditions described above.

The sequence of the state of processes of writing and deletion beginning with state S_0 will be labeled as:

$$\mathbb{S}_{\mathbb{T}} \triangleq \left\langle S_0, S_1, S_2, \ldots, S_T \right\rangle \tag{22}$$

and the set of possible states of this process for the t time period will be labeled as $\mathbb{S}_t$. This way $\mathbb{S}_{\mathbb{T}} \in \mathbb{S}_0 \times \mathbb{S}_1 \times \cdots \times \mathbb{S}_T$. Not all sequences of states are physically possible, as each subsequent condition results from the previous one and the action undertaken. For example, condition (1) must be met regarding the change in time of the values for the deletion counter. Each of those possible sequences can be treated as a trajectory of the changes of the states in the process of writing and deletion.

We shall assume that the process of change in time is ahistorical, which results from the previously made assumptions.

6. Description of the action

Following the previously made assumptions, we shall only discuss two types of actions undertaken by the system user. These are the operations of file writing and deletion.

In the case of file writing, the name of the file to be written is indicated. Let $N_t \in Z(\alpha)$ denote an action undertaken at the beginning of the time period t. Following the assumption about the complete information regarding this file, at the moment of the attempt to write the file, the system knows its size expressed in the number of $c_t \in \mathbb{P}$ clusters.

When the action undertaken is that of file deletion, then only the name of the file to be deleted is given $N_t \in Z(\alpha)$, and all other actions are undertaken by the system.

Thus, a possible action a_t undertaken at the beginning of the t time period can be defined as follows:

$$a_t \triangleq \left\langle b_t, N_t, c_t \right\rangle \tag{23}$$

with

$$b_t \in \{WRITE, DELETE\}, \quad c_t = 0 \ \text{ if } b_t = DELETE \tag{24}$$

The action a_t can be undertaken with the S_t state of the process if this is deletion, or if it is the action of writing a file of a size that allows its writing on clusters of the sets $\mathbb{O}_D(t)$ and $\mathbb{O}_U(t)$. In this situation, the set of possible actions depends on the state of the process. The set of all possible actions is denoted by $\mathbb{A}$. Sets of possible actions for given states are defined by:

$$g : \mathbb{S} \to 2^{\mathbb{A}}. \tag{25}$$

For this function we assume that its value $g(S_t) \subseteq \mathbb{A}$ for each state must not be an empty set, hence $a_t \in g(S_t)$. The sequence of actions of writing and deletion is defined as

$$a = \left\langle a_0, a_1, \ldots, a_{T-1} \right\rangle \in g(S_0) \times g(S_1) \times \cdots \times g(S_{T-1}) \tag{26}$$

This sequence can be interpreted as control of the process of writing of deletion with a corresponding $\mathbb{S}_{\mathbb{T}}$ trajectory defined by formulas in (21) and (22), which runs the process from the initial S_0 state (usually unknown) to the final S_T state (current state). Each transition from the S_t to the S_{t+1} state resulting from performing the $a_t \in g(S_t)$ function is described by the transition function.

7. The transition function

Let us assume that the sets $\mathbb{S}_t$ of physically possible states of processes at the beginning of the t-th time period are identical for all time periods and equal $\mathbb{S}$. Therefore, we look at the process of writing and deletion in stationary conditions, which implies, among others, a constant size of the MD disk, expressed in clusters. In this case, the transition function is determined on the Cartesian product of the set of possible $\mathbb{S}$ states and the set of physically possible actions $\mathbb{A}$:

$$F : \mathbb{S} \times \mathbb{A} \to \mathbb{S} \cup \{IMPOSSIBLE\} \tag{27}$$

with

$$F(S_t,a_t)=\begin{cases}S_{t+1} \ \textit{if} \ a_t \in g(S_t) \\ \textit{IMPOSSIBLE} \ \textit{if} \ a_t \notin g(S_t)\end{cases} \quad (28)$$
$$\textit{for} \ \ t=\overline{0,T-1}$$

Let us further assume, following the prior assumptions, that in the situation where $a_t \notin g(S_t)$, such an action is omitted and the number of time periods reduces by one. Hence, condition (26) is always met. The values of the transition function are constructed as a result of the BFA algorithm and the recording of this result in the $MFT table. The BFA algorithm, working as the first for the action of writing, modifies the content of the disk, that is sets $\mathbb{O}_D(t)$ and $\mathbb{O}_U(t)$, and then a modification of the $MFT table corresponding to these changes is made. However, for the deletion operation, only the $MFT table is modified. This allows an analytical description of the transition function for any process state.

We assumed that the state $S_T \in \mathbb{S}$ is known and we wanted to obtain information about the initial S_0 state. In a generalized case, this is not possible as we do not know the sequence a of actions undertaken. However, we may, based on the knowledge of the transition function F, determine the subset of the $\mathbb{S}$ set of possible states at the beginning of the time period as a compilation of subsequent inverse images of the function F. And so, the set of permissible states at the beginning of the time period numbered $T-1$ is as follows:

$$\overline{\mathbb{S}}_{T-1} \triangleq \left\{S \in \mathbb{S} : \exists a \in g(S) : F(S,a)=S_T\right\} \subseteq \mathbb{S} \quad (29)$$

The set of permissible states at the beginning of the time period numbered $t-1$ is as follows:

$$\overline{\mathbb{S}}_{t-1} \triangleq \left\{S \in \mathbb{S} : \exists a \in g(S) : F(S,a) \in \overline{\mathbb{S}}_t\right\} \subseteq \mathbb{S} \quad (30)$$
$$\textit{for} \ \ t=\overline{1,T-1}.$$

The above recursive definition (29), (30) allows us to designate the $\overline{\mathbb{S}_0}$ set, which incorporates the S_0 state that we are interested in. With additional information about the file or files we are looking for, we can limit this set to its subset and analyze it to discover the lost data.

8. Conclusions

On the basis of the mathematical model presented, it is possible to construct IT tools in the form of an appropriate package of programs for analyzing the existing state of the disk for information about its past condition. This Finite-State-Machine approach was proposed by Gladyshev back in 2004 [5], and even had a real case implementation with some limited success [6]. Since then, this area of research was kind-of abandoned, because of computation complexity [7]. This method of analysis should be modified, especially when there are numerous resulting sets of possible initial disk states. For example, we can analyze the stream of action as a non-uniform Poisson stream and limit ourselves to analyzing the states whose probability of occurrence is greatest. There also exist other useful tools for supporting the analysis, e.g. Markov processes apparatus or control theory methods with the introduction of example control costs for subsequent stages of the process of writing and deletion as costs resulting from data loss.

Information about actions that took place on hard drive could be derived from additional complex analysis of file system metadata. Analysis of NTFS system file $ObjId presented in [8] is one such example. In practice, the specialist often has additional information, e.g. from the system users to which file was deleted, when it was deleted and what writing and deletion operations took place in the analyzed time. Equipped with such additional information, the proposed tool should select one or more areas on the disk where the file may be found. These designated areas can then be examined by a specialist either “manually” or using the methods presented in, e.g. [1].

The proposed method offers an advantage in the form of the ability to recover information about partially overwritten or fragmented files. It also allows us to link the retrieved information with the time of their occurrence. Limiting the search space may allow the specialist to analyze the indicated areas him/herself, which is crucial when searching for files with a specific structure.

9. Bibliography

[1] Darnowski F., Chojnacki A., “Selected Methods of File Carving and Analysis of Digital Storage Media in Computer Forensic”, *Przegląd Teleinformatyczny*, T.3(21) Nr 1-2 (39) 2015.

[2] Darnowski F., Chojnacki A., “Writing and Deleting files on hard drives with NTFS”,

Computer Science and Mathematical Modelling, No. 8, 5–15 (2018).
[3] Ghotge V., Nema P., "Description of the Cluster Preallocation Algorithm in the NTFS File System", *Microsoft Product Support Services White Paper*, 2004.
[4] Carrier B., "File System Forensic Analysis", *Addison Wesley Professional*, New York, 2005.
[5] Gladyshev P., Patel A., "Finite State Machine Approach to Digital Event Reconstruction", *Digital Investigation*, Vol. 1, Issue 2, 130–149 (2004).
[6] Gladyshev P., Patel A., "Finite state machine analysis of a blackmail investigation", *International Journal of Digital Evidence*, Vol. 4, Issue 1, 1–13 (2005).
[7] Olivier M., "Scientific theory of digital forensic", in: *Advances in Digital Forensics XII: 12th IFIP WG 11.9 International Conference*, pp. 3–24, New Delhi, January 4–6, 2016.
[8] Nordvik R., Toolan F., Axelsson S., "Using the object ID index as an investigative approach for NTFS file systems", *Digital Investigation*, Vol. 28, 30–39 (2019).

Model procesu zapisu i kasowania informacji o plikach na dysku z systemem NTFS

A. CHOJNACKI, F. DARNOWSKI

Celem artykułu jest przedstawienie modelu matematycznego procesu zapisu i kasowania informacji o plikach na dysku przy wykorzystaniu zawartości pliku systemowego $MFT, czyli w pliku generowanym w systemie plików NTFS (New Technology File System). Przedstawiony model posługuje się językiem teorii sterowania, utożsamiając stan systemu ze stanem dysku oraz stanem pliku $MFT, a sterowanie jako podjęcie akcji zapisu lub kasowania. Założono przy tym deterministyczny charakter procesu oraz jego stacjonarność. Proponuje się, aby następnie na podstawie funkcji przejścia, po jej uszczegółowieniu, konstruować kolejne przeciwobrazy zbiorów możliwych wcześniej stanów w kolejnych etapach procesu zapisu lub kasowania. Uzyskane rezultaty są podstawą opracowywanych implementacji.

Słowa kluczowe: dysk twardy, NTFS, $MFT.