THE ORGANIZATION OF CRIMINAL IDENTIFICATION INFORMATION

J. F. Phillips – C. J. Whittaker
Canadian Police Information Centre
Royal Canadian Mounted Police

If it were such that the subjects of all criminal record files had their respective file numbers indelibly tattooed on the back of their hand then, with the exception of a few who would chop their hand off, the identification of criminals and subsequent access to the information repository would not be a problem.

As it happens, the law enforcement community is not nearly as fortunate and frequently finds itself with minimal amounts of information upon which it must proceed.

Within this framework it is clearly evident that one cannot discuss personal attribute information organization (data organization) without recourse to how that organized information is itself collectively organized (file organization).

Data organization refers to the actual arrangement of stored data and, file organization refers to the structuring of the arranged data. However, the trouble with terminology – as with data – is that meaning is a function of context. Therefore, we can assume an on-line, multi-tasking, transaction-oriented law enforcement computer system within which a solution is presented.

One of the most difficult problems encountered in the development of the viable law enforcement information system is the search for a data and file organization technique best suited to the magnitude and natural distribution of names and associated index data. The solution so applied must of necessity optimize bulk storage utilization while at the same time not complicating or increasing the speed of information access.

The organization of name-oriented criminal identification data accepts the premise that an individual has or employs some family name – a surname. Consequently, one course of action is to organize relative to these names and accordingly, to organize the supporting name-oriented identification data.

Like many name systems before, this is precisely what has been done!

Embarking initially upon this tack involved an examination of the name distribution within the file. Is a name a prerequisite to criminality? Does the distribution of surnames within the criminal population conform to known distributions of other large non-criminal name files? How is the surname distribution affected by the individual distribution?

To support properly any overall distribution it was necessary to ascertain whether the distribution was stable or was affected by, for example, monthly additions which may alter the structure so that the distribution when totally formulated only reflects its position and composition for that month. If these monthly additive influxes supported the overall distribution, then it is statistically 'safe' to assume that the distribution, when measured, reflects the true picture.

The associated graphs, measured from 1450, 4600 and 5100 monthly record additions, reflect a stable surname initial letter distribution for the periods April 1948 and 1968 as well as August 1968 and individually reflect and equally support a total file distribution throughout 2.3 million name records.

Relative to the initial question now, the absence of name criminality is reflected in and is evident from the comparison of the total criminal file distribution to a "control" distribution file of a one-million individual, 144.5 thousand surname pension file. Group sizes are not contingent upon the surname distribution but on the individual distribution within each surname initial letter group – there may be more surnames whose initial letter begins with M but there are more individuals encompassed by the S-surname initial letter grouping.

If name files and their subsequent accesses were equally distributed across the surname groupings and the associated number of individuals involved then, any discussions relative to groupings and group sizes would be superfluous.

Further compounding the erratic nature of such a file is the fact that computers are not heuristic and as usual rely on a finite algorithmic approach to problem solving.

Names by definition present a problem! Their structure, their use and, their sound are all functions of their environment. Have they been transliterated? Are they formally documented? Is this a criminal environment where the individual motiva-

tion for identity concealment is high? Was the name in question merely hearsay and represents only a phonetic translation?

Embedded within any algorithmic approach to the name-handling problem should be capabilities to accommodate the multiplicity and combination of yes/no answers just to the four aforementioned situations.

As our own experience indicated, one of the reasons why the Soundex and Soundex-type versions of name codification has been widely accepted and employed, primarily in manually-oriented environments, is not only because they provide sound-equivalent classes of proper names but their inherent capabilities lie within the realm of organizational smoothing which compensates for the unevenness of name distributions.

However, with subjectivity removed as a consequence of manual-by-computer replacement, especially within a criminally-oriented environment, additional emphasis must be placed on the codification technique employed to compensate for individual subjectivity inherent within the employment of Soundex-type and code-to-letter-type algorithmic approaches. By necessity, subjective considerations must be algorithmically included to compensate for:

1. the de-emphasis of the initial surname letter within the respective name code;
2. excluding the codification of silent, intrusive consonants;
3. the problems of the ethnic disparity of names within the overall population;
4. phonetic discontinuities relative to the same name within different geographic areas, and,
5. to compensate for, especially in the criminal environment, increased motivation towards identity concealment.

Now, since the name within this structure is the key, attempts must be directed toward the integrated solution of the above considerations within an objective algorithmic technique. Consequently, it was necessary also to try to take advantage of the ensuing, and as usual, erratic distribution.

Rather than approaching these two problems – name codification and organization – from a single-solution standpoint; knowing that something would suffer, we effected a dual approach with the initial emphasis on name codification. Its solution was not to be dependent upon any preconceived notion of smooth organizations, 'flat' curves or equal overall distributions. The secondary problem, as previously stated, was to organize the file and its associated data within the resultant distribution in such a way as to optimize bulk storage and minimize file access.

Most name-grouping techniques in operation or available today, automated or manual, are primarily algorithmic and are designed to accommodate name alterations within the following areas:

1. spelling (orthographical);
2. sound (phonological);
3. spelling/sound;
4. structure (morphological);
5. substitutions by new and/or linguistically unrelated names (lexicographical).

To this end, relative to our own peculiar context and environment, we have developed a *ph*onetically *i*ndexed *n*ame *d*irectory code (FIND) as a solution to the initial problem area. Primary emphasis in the development of the FIND technique was directed towards the solution of problems which were pertinent to categories (1), (2) and (3). Grouping and subsequent searching relative to the latter two categories for the most part, relies on linguistic subjective intervention. In all of these categories there exists sub-groups which involve further sound and/or spelling alteration. Until such time as computers become heuristic, there is very little which can be done to compensate for the deliberate name abbreviations, combinations and hybridizations.

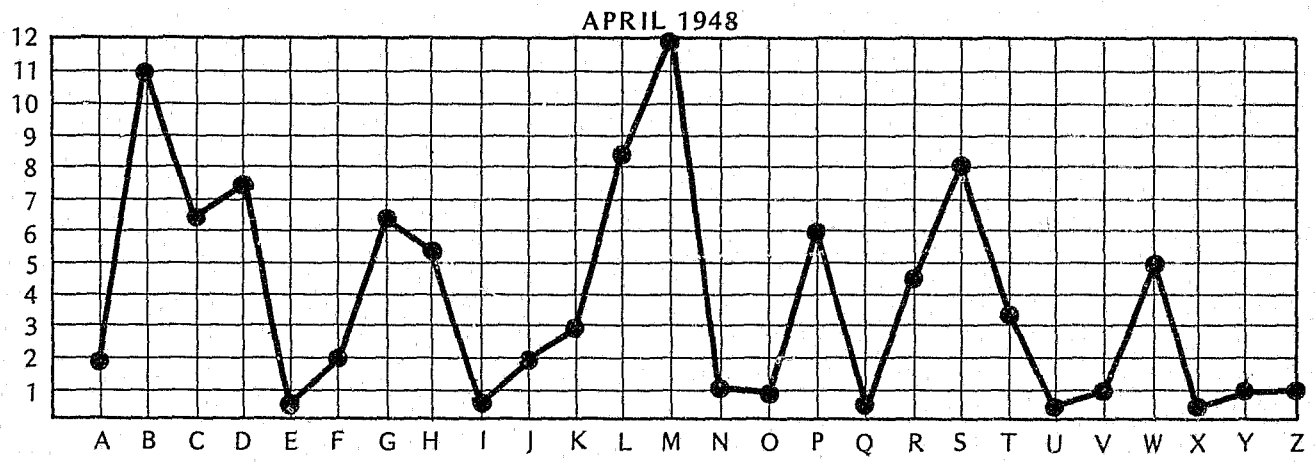
Within our approach we have deliberately increased our emphasis towards the first three categories because of the context within which the names are used. In North America, it is usually the environment which influences the phonological and ultimately the orthographic structure of a name. Usually, names of different ethnic origins have been transliterated in an anglicized sense and as such are relatively 'easy' to accommodate.

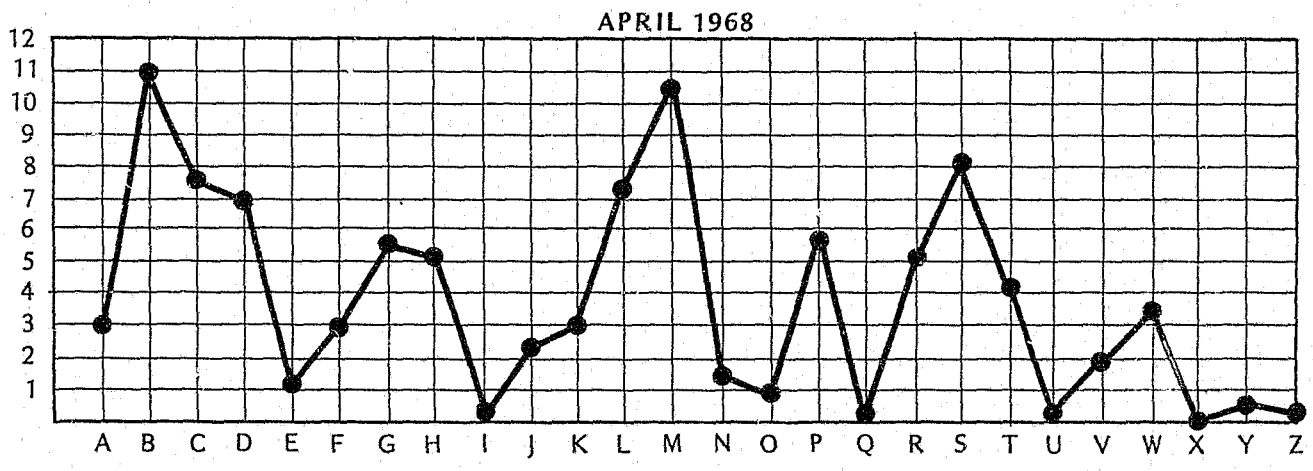
However, coupled with the ethnic disparity throughout Canada we have also a bilingual bicultural national complexion. As a result, the phonological adjustment of names is not always evident. To compensate for this type of phonetic translation, as well as orthographic transliteration, we have included within the structuring of the FIND technique a mechanism of cross-referencing.

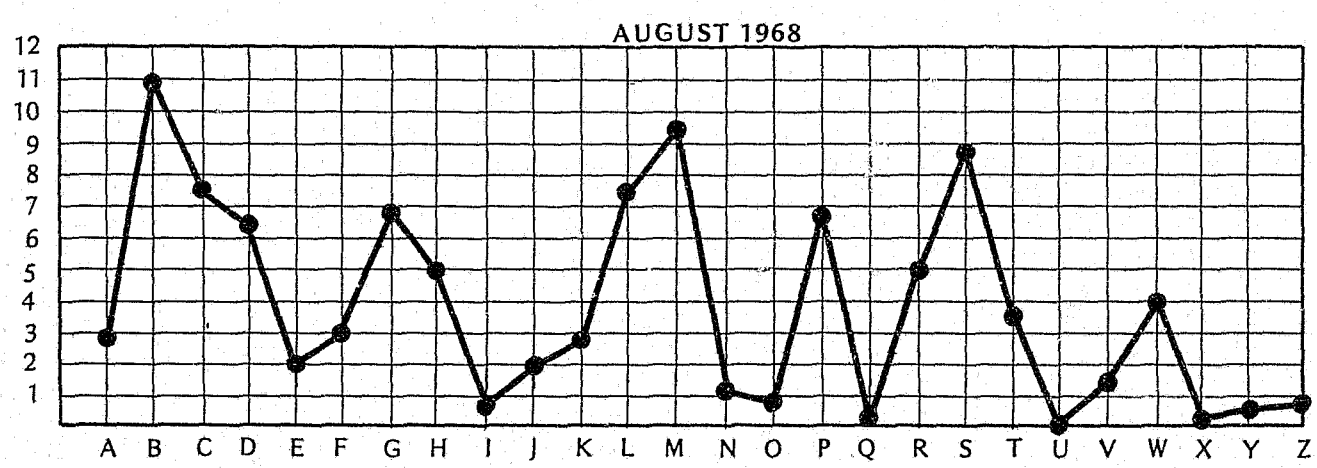
To be more specific and to cite just one particular letter grouping examine the structure context of 'GH'. It may be silent or have either a K/G or F sound but, the silent sound in one ethnic environment may become an F-sound in another environment. Likewise, the F-sound may be a K or G sound in a different geographical area.

In Canada, to a French-speaking individual the names GOYER and BARIL regardless of their origin or ownership are pronounced GOY-AYE and BAR-REE whereas to an English-speaking individual the name may be pronounced as before and as GOY-ER and BAR-IL.

PERCENTAGE OF TOTAL ENTRY VS SURNAME INITIAL

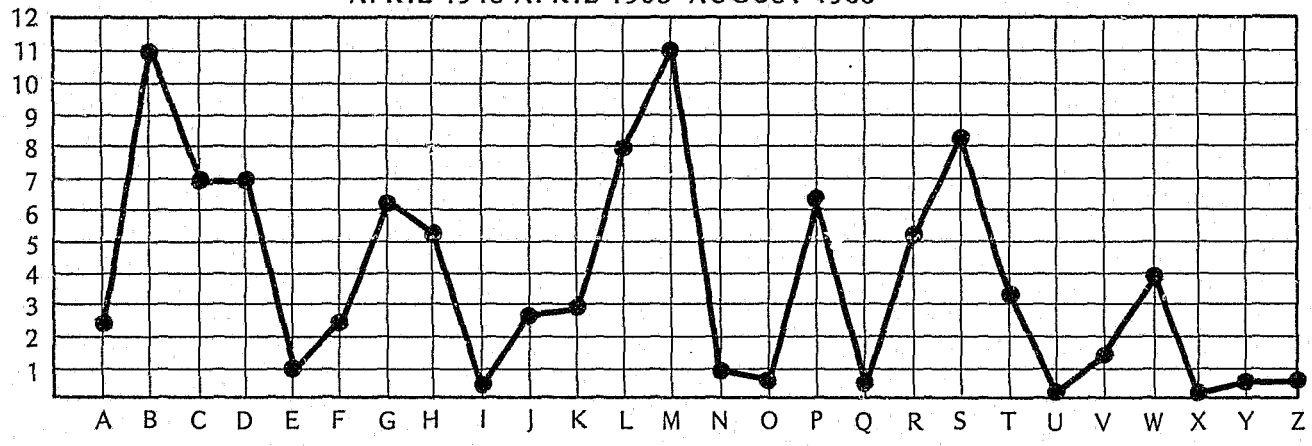




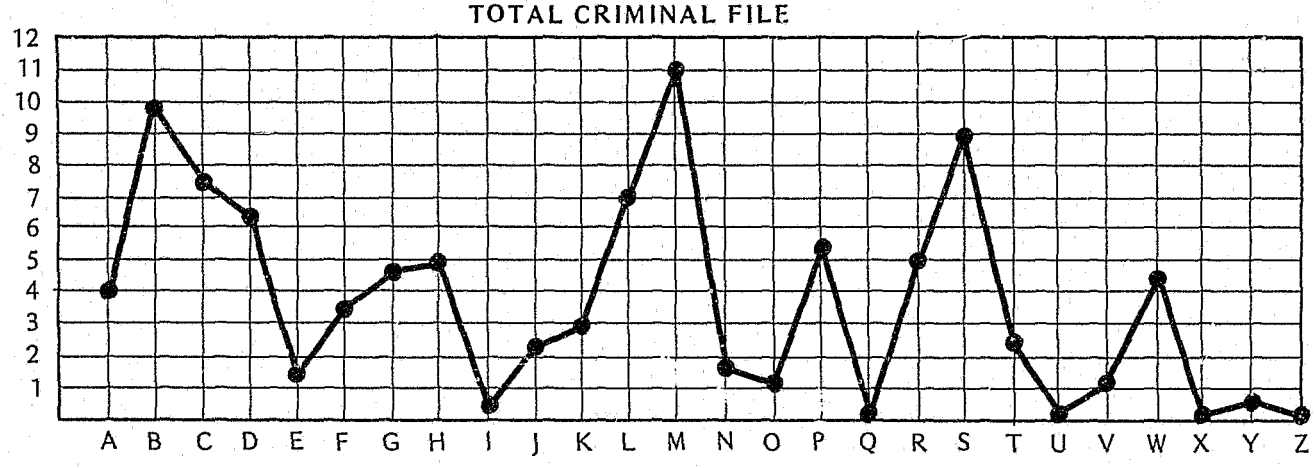


PERCENTAGE OF TOTAL ENTRY VS SURNAME INITIAL

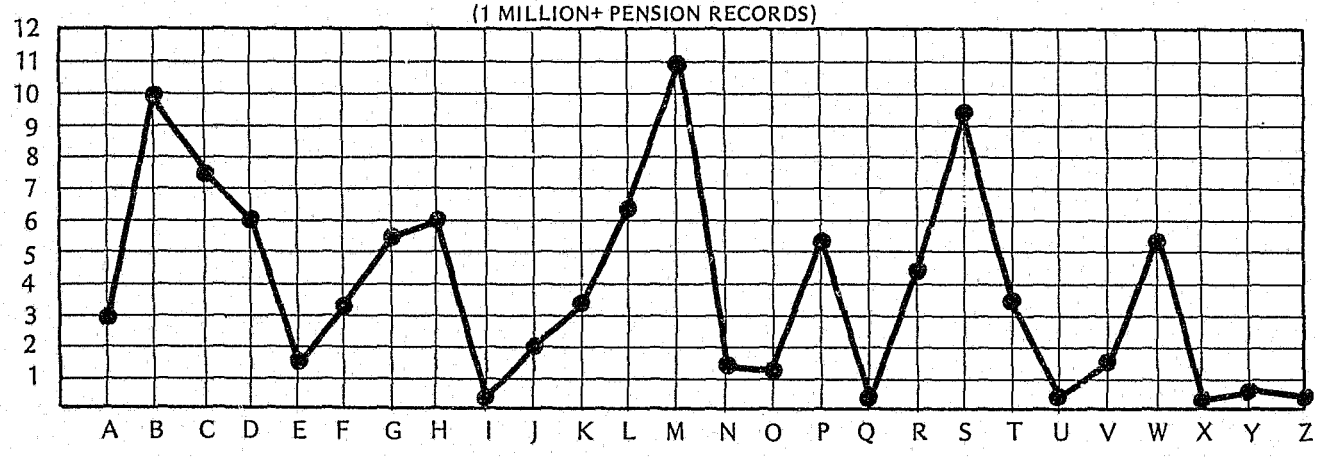
APRIL 1948 APRIL 1968 AUGUST 1968



TOTAL CRIMINAL FILE



NON-CRIMINAL CONTROL FILE
(1 MILLION+ PENSION RECORDS)



Other names, dependent upon their environment and their ownership, fall into a similar category. The following names and their derivatives are shown to elucidate the FIND technique and its automatic cross-referencing.

NAME/FIND CODE, *CROSS REFERENCE FIND CODE	"USUAL" PRONOUNCIATION	NAME DERIVATIVES		
DAILLEBOUST	DIEBO	DALIBOST	DOWBUSH	DALEBOUGH
D8009	D8009			
*D5860		D5860		
*D8600			D8600	
*D5809				D5809
				*D5870 (DALEBOK)
				*D5880 (DALEBUFF)
MAUGHAM	MAWM	MOAKAM	MOFFAM	
M2000	M2000			
*M7200		M7200		
*M8200			M8200	
LEFEBVRE	LEFAIVE	LEFEEVER	LEFAIVRE	
L8800	L8800		L8800	
*L8840		L8840	*L8840	
BELVOIR	BELLEVOIR	BEAVER		
B5840	B5840			
*B8400		B8400		

SOME INTERESTING PHONETICALLY EQUIVALENT BUT ORTHOGRAPHICALLY DISSIMILAR FIND-GROUPED NAMES

CARLYLE	CARLISLE
DIXON	DICKSON
DAVISON	DAVIDSON
EWEL	YULLE
FOYTOE	FEUILTAULT
HOMES	HOLMES, OHMS
JOHNSON	JOHNSTON
LAWRENCE	LORENTZ
LEVINE	LAVIGNE
LEE	LEIGH
LESTER	LEICESTER
LECLARE	LECLERC
MAYER	MEILLEUR
PORSHES	PORTEOUS
RISHOVA	HRYCZOWA
SEAR	CYR
THOMSON	THOMPSON
WANG	HUANG
YOUNG	JUNG

The search for the file organization method which was best suited to the inherent characteristics of the criminal file was not without obstructions. The most significant of these which had to be and were surmounted was, as previously stated, the ungovernable and erratic distribution effected by the commonality of surnames. Since the distribution of the FIND code is directly proportional to that of surnames, any attempt to improve upon this distribution by "organizational smoothing" would jeopardize the phonetic structure of the FIND code.

The next characteristic in order of significance which proved to be as much of a benefit as a perplexity, was the highly repetitive nature of the file content. Additional characteristics of a lesser degree of significance were presented by the need for direct access by FIND code, optimal disk space utilization, and optimal read times.

There are two basic groups of inherent characteristics which can be applied to any file. They are data file characteristics and processing characteristics.

Data file characteristics can be grouped under volatility, activity and size. With respect to the criminal name index file, relative to the 2.5 million on-line records, Figure 1 provides a list of the fields of data to be associated with each record. The amount of data if stored in display format totals 87 bytes per record, or in an optimized multi-data type structure may be stored within 66 bytes. However, further savings have been achieved through the use of codes and multi-level file structure.

The criminal name index file is not highly volatile. Indications to date show an annual gross

increase of approximately six percent accompanied by an annual gross decrease of approximately one percent giving a net increase of five percent.

Activity, as a data file characteristic, takes on a different realm of concern when the file under consideration is to serve the purposes of an on-line system. To put it simply, as the degree of file organization efficiency rises, so does the overall degree of system performance and, inevitably, so does the activity.

DATA FIELD	*NUMBER OF BYTES IN DISPLAY*	*OPTIMIZED BYTE COUNT*	*STORED IN ABB. FORM*
FIND CODE	5	4	NO
SURNAME	24	24	NO
CHRISTIAN NAME (1)	10	4	YES
CHRISTIAN NAME (2)	10	4	YES
DATE OF BIRTH	6	6	NO
PLACE OF BIRTH	1	1	YES
HEIGHT	3	2	NO
WEIGHT	3	2	NO
COLOUR OF EYES	1	1	YES
SEX	1	*	YES
RACE	1	*	YES
FINGERPRINT PATTERN TYPE	10	10	YES
PRIMARY FPT. CLASSIFICATION	4	2	NO
SECONDARY FPT. CLASSIFICATION	2	2	NO
FILE NUMBER	7	4	NO

* These fields are stored in combination with other fields: Female Sex = Day of Birth +50; Coloured Race = Height (Ins.) +100.

FIGURE 1

The primary processing characteristic is that the processing of the search against a group cannot of necessity be terminated even in the case of achieving a complete match to all criteria provided. Therefore, there is nothing to be gained by providing direct access to any particular location in a group.

The file organization method selected for the criminal name index must provide for the direct access to all individuals, on a group basis, whose surname pertains to a common FIND code. Therefore, the distribution of the file is subject to the same distribution as that of the FIND code. Since the distribution of the FIND code is directly proportional to that of the surnames, any attempt to improve upon this distribution by 'organizational smoothing' would jeopardize the phonetic structure of the FIND code.

The solution exists within a file organization method which utilizes two levels of data storage (header and detail) and file distribution of detailed data (across sub-files of different page sizes) in accordance with FIND code group sizes thus employing the aforementioned erratic distribution for the most efficient utilization of direct access resources (97% utilization factor as opposed to 73% with a fixed page size).

Information retrieval is via the header file to the detail file. Direct access to the header file is effected by a radix transformation of the binary coded FIND code. The use of the radix transformation randomizing technique for the organization of the header file provided three advantages:

1. it furnished the necessary key for the direct access to a FIND code group;
2. it combined surnames of multiple FIND code groups in such a manner as to provide an even distribution for the optimizing of disk space utilization (i.e., 86% in the header file, and 97% in the detail file;
3. the number of individuals represented are equally distributed among the 1,000 transformed radix values.

The dual-level file concept was employed for many reasons, but among the more predominant was that it provided a means of storing the whole surname only once for each unique surname within the file.

A surname number unique within a radix value was assigned to each surname and was stored within the respective detail record as an identifier of the exact surname spelling for use in record matching and descriptor weighting. As an aid to the optimizing of read times, the surname records within a transformed radix value were stored in sequence of their frequency of occurrence. In spite of the average of 250 surnames per radix value with only 41 to a physical sector (page) of disk space, this sequencing of surnames provided for a match on a surname within the first of many pages (six or more) for 80% of all enquiries.

FILE DESIGN AND RECORD STRUCTURE

HEADER FILE

The header file has a page size of 41 records and a record size of 40 bytes and contains one entry for each surname within a FIND code. Each record contains the full surname with its respective data including the relative record number of the first detail file page for the respective FIND code.

The pages are organized by means of a radix transformation of the FIND code (Figure 3).

The entries pertaining to each transformed radix value are sequenced in descending order relative to their frequency of occurrence within the detail file. No attempt has been made to maintain this sequence of entries due to its relatively static nature. The layout of the header file is displayed in Figure 2 with a sample page in Figure 2A.

HEADER FILE ENTRY LAYOUT

FIELD NAME	*DESCRIPTION*
SURNAME	First characters of surname, no imbedded spaces or special characters;
REL-REC-NUMBER	Relative record number of first detail page for respective FIND code;
FILE-NUMBER	File number of detail sub-file for respective FIND code;
SURNAME-NUMBER	Surname number (relative to a FIND code) used to represent exact spelling of surname within detail file;
SURNAME-COUNT	Current count of individuals in index with surname as spelled within surname;
FIND-CODE-COUNT	Current count of individuals in index with respective FIND code;
FIND-CODE	FIND code for surname as stored in surname.

FIGURE 2

SAMPLE HEADER PAGE

MacDonald	41800	19	0001	3048	4669	M7125
McDonald			0002	1390		M7125
Gamache	22780	19	0003	130	334	J2600
Genest			0004	102		J2600
Melnychuk	38860	19	0005	56	151	M2670
McDonell			0006	56		M7125
MacDonell			0007	51		M7125
MacDonnell			0008	51		M7125
McDonnell			0009	48		M7125
Borland	2280	19	0010	39	102	B4521
Blundell	3559	19	0011	35	55	B5250
Janez			0012	22		J2600
Starley	52460	19	0013	17	121	S1479
McDaniel			0014	16		M7125
Swiderski			0015	16		S1479
Berlinguette			0016	15		B4521
Breland			0017	13		B4521
Navratil	584	18	0018	12	13	N8150
Mandziuk			0019	11		M2670
Melnichuk			0020	9		M2670
Dhaliwal	528	17	0021	8	9	T5500
Brillant			0022	7		B4521
Burlington			0023	7		B4521
Yanish			0024	7		J2600
Mensch			0025	7		M2670
Nattrass	43350	19	0026	7	34	N1409
Gionest			0027	6		J2600
Boreland			0028	5		B4521
Ballingall			0029	5		B5250
Janisch			0030	5		J2600
Manchuk			0031	5		M2670
Mandzuk			0032	5		M2670
Niddrie			0033	5		N1409
Straka			0034	5		S1479
Zatorski			0035	5		S1479
Dengler	53	17	0036	4	10	D2540
Odonaughy	80	17	0037	4	8	E1289
Odonoughue			0038	4		E1289
Gunst			0039	4		J2600
Manske			0040	4		M2670
Monasch			0041	4		M2670

FIGURE 2A

The content of the sample header page indicates that there exists multiple header page(s) pertinent to this example. All 41 entries within this page have been used and, page multiplicity is indicated by the fact that the surname count at the 41st entry encompasses four or less individuals. Consequently, no attempt should be made to reconcile the accumulation of all surname count entries for each FIND code entry with that of its corresponding FIND code count-entry.

For example, the FIND code count entry with FIND code M7125 = 4,669 whereas, the sum of all surname counts for FIND Code M7125 (i.e., at 1, 2, 6, 7, 8, 9 & 14) = 4,660, nine less. Therefore,

there exists at least two more surnames each having four individuals and one surname with one individual at the minimum number of additional surname level to a maximum of 9 surnames each having only one associated individual thus accounting for the nine-individual difference.

RADIX TRANSFORMATION

The FIND code for the surname JOHNSON is J2620. This FIND code is transformed to a different radix or base, excess digits are discarded leaving a relative record number of the required length.

An example of FIND code J2620 conversion to radix 11 to produce a relative record number for the respective header file page is shown in Figure 3.

RADIX TRANSFORMATION

STEP 1: CONVERT FIRST CHARACTER OF FIND-CODE TO NUMERIC VIA THE FOLLOWING TABLE:

B	0	F	3	L	6	P	9	T	12
D	1	J	4	M	7	R	10	U	13
E	2	K	5	N	8	S	11	W	14

J2620 = 42620

STEP 2: $(4 \times 11^4) + (2 \times 11^4) + (6 \times 11^3) + (2 \times 11^2) + (0 \times 11^1) = 644204 + 29282 + 7986 + 242 + 0 = 681714$

STEP 3: ADD 1 TO LAST THREE DIGITS TO ELIMINATE THE POSSIBILITY OF ZERO, 999 WOULD BECOME 1000.
714 + 1 = 715

STEP 4: SINCE 6 PAGES OF 41 ENTRIES ARE REQUIRED FOR EACH RADIX VALUE, MULTIPLY THE RESULT OF STEP 3 BY 6 AND SUBTRACT 5 TO OBTAIN THE FIRST RELATIVE RECORD NUMBER FOR THE TRANSFORMED RADIX VALUE.

(715 x 6) - 5 = 4290 - 5 = 4285

RELATIVE RECORD NUMBER = 4285

FIGURE 3

DETAIL FILE

The detailed file level of the criminal index file structure is comprised of multiple sub-files. The only structural difference between all of these sub-files is the number of entries per page.

FIND code groups have been allocated to the various sub-files as determined by the number of individuals within the FIND code. The file number and relative record number for the first page of each FIND code is stored within the header records for that FIND code.

The sequence of entries within a group of pages for a FIND code were, at time of file creation, by number, but no attempt has been made to maintain this sequence as it is of no consequence to the efficiency of the system. The layout of the detail sub-files is displayed in Figure 4 and a sample page in Figure 4A.

DETAIL FILE ENTRY LAYOUT

FIELD NAME		*DESCRIPTION*
FP-NUMBER	0	Fingerprint section file number. Full word binary;
SURNAME-NUMBER	4	Represents exact spelling of surname as stored within header file. Half word binary format;
PRIMARY	6	Primary fingerprint classification of subject. Binary format;
SECONDARY	8	Secondary fingerprint classification of subject. Display mode.
GIVEN1	10	First given name of subject; First 4 characters (equivalences JACK = JOHN etc)
GIVEN2	14	Second given name of subject First 4 characters;
DAY	18	Day of birth of subject; (plus 50 if female)
MONTH	20	Month of birth of subject;
YEAR	22	Year of birth of subject;
BIRTHPLACE	24	Birthplace code;
EYE-COLOUR	25	Eye-colour code;
HEIGHT	26	Contains height of subject in inches plus 100 if race is non-white. Packed decimal.
WEIGHT	28	Contains weight of subject in pounds.
PATTERN-TYPE	30	Contains 10 fingerprint pattern types proceeding from right thumb to left little finger. Display mode.

FIGURE 4

SAMPLE DETAIL PAGE

024660	0001	2428	WW	GEO	D	21	02	99	1	1	60	125	WWDRTWWCDX
040703	0001			MIC	RIC			01	0	1	61	135	RWW=DDCCDT
042756	0002	1022	TT	TOM	J			00	2	2	59	120	WTTWCUTTXD
056222	0001	3228	DW	WIL	ROB	21	04	02	2	4	63	129	CDXWAWWWWW
059426	0006	2303	XW	HARR	HEWI	21	08	97	1	0	65	137	AXTUUWWCTT
070094	0001	2630	CT	JOE	CAL	23	12	03	3	1	64	160	WCDWWTTACX
099715	0001	0325	TR	DAV	LES	14	06	10	0	2	60	150	CTDARARCUA
137989	0002			GEO	MAUR	27	03	99	4	1	67	149	DAWX=TUWXW
147229	0001			HARR	A	15	12	09	2	1	69	165	XRWTW=TWCD
159728	0001	0720	AW	JAM	DAV	21	15	06	3	4	68	166	DATRAWWXDA
168206	0009	0126	TT	CEN	GERA	09	07	03	2	0	160	095	WTWUUTTUWR
173308	0001	0102	AT	MILD		55	01	06	1	2	68	152	AARRAATTDT
185854	0014	1807	DX	ART	J	20	03	97	3	2	69	165	RDAUWTXRAW
219085	0001	2614	XR	GEO	JOE	29	08	99	8	1	66	192	AXCCARRRDD
232889	0001	1313	UT	JOH	JOE			06	9	1	67	158	TUWCXWTAAA
268194	0002			JOE	LEW	30	12	14	3	1	68	201	U==CUTATDC
387639	0001	3130	WR	ROB	ARCH	15	10	16	6	4	67	169	CWDDXDPWDT
463875	0007	2830	WT	DEN		01	12	17	4	0	69	175	WWWDCATCXW
586290	0001	2431	DX	JOE	MAUR	16	02	27	3	2	72	195	CDXACCXWUD
611479	0001	1126	TA	AL		03	05	23	7	1	71	185	WTCDRUAWDA
675655	0008	0108	AW	FRED	WIL	19	08	17	2	0	67	140	AARTDAWUWR
689842	0001	3239	XU	DON	LAWR	03	02	24	3	2	66	157	DXWWCCUWWX
698377	0001	2531	XW	JAM	RALP	06	01	29	0	2	69	180	CXWCDRWTRT
724361	0002			JOH	HENR	25	01	22	1	1	70	165	XCDRAXXUA=
759651	0001	0801	TT	MURC		04	10	35	3	1	165	90	TTAUADTWAX
812520	0001	1824	WD	LAWR		06	02	36	2	0	69	159	XWAUXRDACW
821922	0006	2025	WT	STEW		23	06	39	4	0	72	215	WWDXAATPTW
872020	0001			CHAR	CLAY	01	12	40	3	4	74	225	WCTUWRART=
901928	0007	3224	WC	GORD	WALT	01	01	36	2	1	69	139	CWTWWCCDXD
942034	0001	0604	TC	LAWR	MING	02	06	20	2	1	68	165	ATUUAWCUWD
996129	0002	1205	RU	MAUP	JOE	20	06	41	3	1	72	215	ARUWWUUDTC
1050140	0001	1308	RW	CHAR	DAN	26	12	41	2	4	69	149	RRTWCCWTXR
1080783	0001	2307	XD	DAV	GLEN	28	07	44	3	0	70	195	TXRUCDDCRA
1105667	0008	2316	WX	GEO	DON	25	07	37	2	2	69	187	UWCAWDXXDA
1141971	0001	1431	AX	DON	JAM	20	03	43	6	1	70	176	WADWWCXARX
1218345	0001	0123	RD	MARG	ELIZ	52	03	42	9	1	166	98	CRTUXUDART
1283198	0009	1521	AT	DAV	GERA	08	12	48	4	4	68	159	DATCDCTXRT
1352925	0006	2519	XW	ED	FARL	02	11	52	1	1	68	205	XXRDTUWRAT
1358461	0001	1127	UW	ROB	AL	06	01	49	8	0	71	190	XUDXRUWXAP
1372255	0001	2818	WA	HELE	IREN	70	08	51	3	2	70	180	DWACTRADCW
1376603	0002	1717	WA	NORM	GEO	14	11	51	2	1	72	210	CWUTRAARTU

P/S FINGERPRINT CLASSIFICATION IS NOT RETAINED IF THE
CORRESPONDING PATTERN-TYPE IS UNKNOWN: SCARRED OR AMPUTATED STORED (=)

FIGURE 4A

The evaluation procedures which embody the aforementioned record structures and led to the selection of the file organization technique included a study to facilitate the selection of an optimum record size and subsequent page size.

Across the horizontal axis of Table 1, entry or record sizes are recorded from 24 to 47 bytes. The vertical axis is a count of the number of records per page and has a range of from 1 record per page to a maximum number of records which can be stored within a disk track for the minimum (horizontal) record size. The elements of the table contain the number of pages relative to the respective record-count and record size which can be accommodated within a physical track of disk space. An asterisk element indicates that the space which would be reserved for the respective record-count and size is sufficient to accommodate either an additional record or an additional character within each entry. As such, it indicates that this point does not provide for the optimal utilization of space.

The record counts recorded in Figure 5 are optimal for a record size of 40 bytes. They are the only record counts which were used by the criminal index system as all others resulted in unnecessary amounts of unused disk space.

TABLE 1

LOGICAL INDEX RECORDS/TRACK TABLE

ENT. SIZE	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
1 ENTRIES	120	*	*	*	*	*	*	*	104	*	*	*	*	*	*	*	92	*	*	*	*	*	*	*
2 ENTRIES	84	*	*	*	76	*	*	*	68	*	*	*	64	*	*	*	60	*	*	*	56	*	*	*
3 ENTRIES	64	*	60	*	*	56	*	*	52	*	48	*	*	*	*	*	44	*	*	*	*	40	*	*
4 ENTRIES	52	*	48	*	*	*	44	*	*	*	40	*	*	*	36	*	*	*	*	*	32	*	*	*
5 ENTRIES	44	*	*	40	*	*	36	*	*	*	*	32	*	*	*	*	*	28	*	*	*	*	*	*
6 ENTRIES	*	36	*	*	*	32	*	*	*	*	28	*	*	*	*	*	*	24	*	*	*	*	*	*
7 ENTRIES	*	32	*	*	*	28	*	*	*	*	*	24	*	*	*	*	*	*	*	20	*	*	*	*
8 ENTRIES	*	*	28	*	*	*	*	24	*	*	*	*	*	*	20	*	*	*	*	*	*	*	*	*
9 ENTRIES	*	*	*	24	*	*	*	*	*	20	*	*	*	*	*	*	*	*	16	*	*	*	*	*
10 ENTRIES	24	*	*	*	*	*	20	*	*	*	*	*	*	*	16	*	*	*	*	*	*	*	*	*
11 ENTRIES	*	*	*	20	*	*	*	*	*	*	16	*	*	*	*	*	*	*	*	*	*	*	*	*
12 ENTRIES	*	20	*	*	*	*	*	*	16	*	*	*	*	*	*	*	*	*	*	*	12	*	*	*
13 ENTRIES	*	*	*	*	*	16	*	*	*	*	*	*	*	*	*	*	12	*	*	*	*	*	*	*
14 ENTRIES	*	*	*	16	*	*	*	*	*	*	*	*	12	*	*	*	*	*	*	*	*	*	*	*
15 ENTRIES	*	16	*	*	*	*	*	*	*	*	*	12	*	*	*	*	*	*	*	*	*	*	*	*
16 ENTRIES	16	*	*	*	*	*	*	*	*	12	*	*	*	*	*	*	*	*	*	*	*	*	*	*
17 ENTRIES	*	*	*	*	*	*	*	12	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	8
18 ENTRIES	*	*	*	*	*	12	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	8	*	*
19 ENTRIES	*	*	*	12	*	*	*	*	*	*	*	*	*	*	*	*	*	*	8	*	*	*	*	*
20 ENTRIES	*	*	12	*	*	*	*	*	*	*	*	*	*	*	*	*	8	*	*	*	*	*	*	*
21 ENTRIES	*	12	*	*	*	*	*	*	*	*	*	*	*	*	8	*	*	*	*	*	*	*	*	*
22 ENTRIES	12	*	*	*	*	*	*	*	*	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*
23 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*
24 ENTRIES	*	*	*	*	*	*	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*
25 ENTRIES	*	*	*	*	*	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
26 ENTRIES	*	*	*	*	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
27 ENTRIES	*	*	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
28 ENTRIES	*	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
29 ENTRIES	*	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
30 ENTRIES	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
31 ENTRIES	*	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
32 ENTRIES	*	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
33 ENTRIES	8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
34 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
35 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4
36 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*
37 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*
38 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*
39 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*
40 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*
41 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*
42 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*
43 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*
44 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*
45 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*
46 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*
47 ENTRIES	*	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*
48 ENTRIES	*	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*
49 ENTRIES	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*
50 ENTRIES	*	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*
51 ENTRIES	*	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
52 ENTRIES	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
53 ENTRIES	*	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
54 ENTRIES	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
55 ENTRIES	*	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
56 ENTRIES	*	*	*	*	*	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

280 ETC.

RECORD SIZE: 40 BYTES

OPTIMAL RECORD COUNTS	ENTRY SIZE UTILIZATION FACTORS*
1	40/40 (1.0)
2	40/40 (1.0)
3	40/40 (1.0)
4	40/44 (0.91)
5	40/41 (0.976)
6	40/41 (0.976)
7	40/43 (0.93)
9	40/42 (0.953)
13	40/40 (1.0)
20	40/40 (1.0)
41	40/40 (1.0)

* — Proportion of bytes used to bytes available.

FIGURE 5

Further analysis with respect to measures of central tendency provided a sound basis for estimating the sizes of groups to be processed. The most commonly used measure of central tendency is a mean or average. Accordingly, there are different types of averages and their misuse (and statistics in general for that matter) are indeed responsible for critical errors in judgment. The most commonly used average is the arithmetic mean, the total number of records within the detail file divided by the total number of groups:

$$\text{i.e., } X = 2583232 \div 22250$$
$$= 116.1$$

This value, in all its accuracy, is highly deceptive. It has a value only were it such that all groups, regardless of their size, shared an equal probability of being encountered. However, this is not true and, the more common a surname, the higher the probability of its occurrence in an enquiry. Consequently, a second kind of average was employed and is the weighted mean:

$$M = \Sigma P_i C_i = 2162$$

where P_i = probability of a group size being encountered,

and C_i = the size of the group.

This average is the most appropriate because its formula employs relative weights to each group size within the file proportionate to the number of individuals occurring in the respective group sizes. These weights are directly equivalent to the probability of the various group sizes being encountered by any inquiry.

An additional statistic which is indicative of central tendency is the median. Herein, the median frequency has a class width of 1201 to 1225 individuals demonstrating that 50% of the enquiries will be directed towards groups which have less than 1213 individuals and the remaining 50% to groups which have more than 1213.

FILE UTILIZATION

The detail file now consists of 9 sub-files each having one of the following page sizes: 2, 4, 5, 6, 7, 9, 13, 20 and 41.

The principal reasons for the creation of detailed sub-files, as opposed to one detail file with a fixed page-size, were increased file utilization and optimized read times. This fact is appreciated when comparing the overall disk space utilization of 97.9% provided by the former as opposed to 73.9% achieved through the use of the latter.

READ TIME

All individuals within a particular FIND code must be evaluated in the processing of an enquiry. It has been found that the number of individuals whose surname generated the same FIND code may vary from one to in excess of 14,000. This being the case, it was necessary to be able to vary the size of the FIND code group. This process is available and is achieved through the use of multiple sector reads.

Due to the organization by frequency of occurrence within header file, access is one sector at a time. Similarly, 8 of the 9 sub-files within the detail file contain groups which do not exceed one sector of contiguous disk space. Therefore, the only file which utilizes the multiple sector reading facility accounts for in excess of 97% of all enquiries — the 41 record per page detail sub-file.

The following report indicates the total number of references required in the processing of one thousand enquiries by sector count based upon the distribution of the file of 2.5 million records. The group sizes within this file were examined to determine the number of sectors to be read and the respective entry within the attached table was incremented. The timings include an allowance for track and cylinder overflow.

Had we not opted to use the multiple sector read facility, the resulting average read time per enquiry would bear no semblance to that contained herein. To demonstrate—the timing per reference for an 8 sector multiple read is 125.8 milliseconds, whereas the same 8 sectors accessed

individually, would require 8 times 78.75 or, 630 milliseconds.

READ SUMMARY BY SECTOR COUNT

SECTOR COUNT	REFERENCE FREQUENCY/ 1000 ENQUIRIES (f)	TIME/ REFERENCE (t) (millisecs	f.t
1	171	78.75	13466.25
2	139	85.47	11880.33
3	121	92.19	11154.99
4	114	98.90	11274.60
5	119	105.61	12567.59
6	88	112.33	9885.04
7	138	119.04	16427.52
8	5537	125.80	696554.60

AVG. READ TIME/ENQUIRY ..... *783.21 ms.

*With 1 enquiry/sec., for every hour of CPU time, there is concurrent read time of 46 min. 59.4 secs.

On a single sector read basis, our average read time would have increased by 477% from 783.2 ms to 3,740.7 ms such that the ensuing total concurrent read time with one enquiry per second for every hour of CPU time would increase by the same percentage from 46 mins. 59 secs. to 3 hrs. 44 mins. 26 secs.

As such, in this instance, in its simplest form, the last enquiry output would be forthcoming 3 hrs. and 44 mins. later.

With respect to the header file, it has been determined, and fortunately so, that the distribution of surnames as a consequence of radix transformation, closely approximates the 'normal distribution curve' with an average group size of 248 surnames and a standard deviation of 20.8.

In order to provide for direct accessing of header pages, as determined by the surname within an enquiry, it is necessary that the relative record number for that record be directly calculable from the respective FIND code. Such being the case, the same number of contiguous pages must be allotted to each radix value. For the most efficient utilization of disk space as well as processing efficiency, sufficient pages must be allotted to each radix value so as to accommodate the mean group size. As previously indicated, the record size selected for the criminal index header file is 40 bytes with a respective page-size of 41 entries, with 6 pages of 41 entries (246) reserved within the prime area for each of the 1,000 (transformed) radix values.

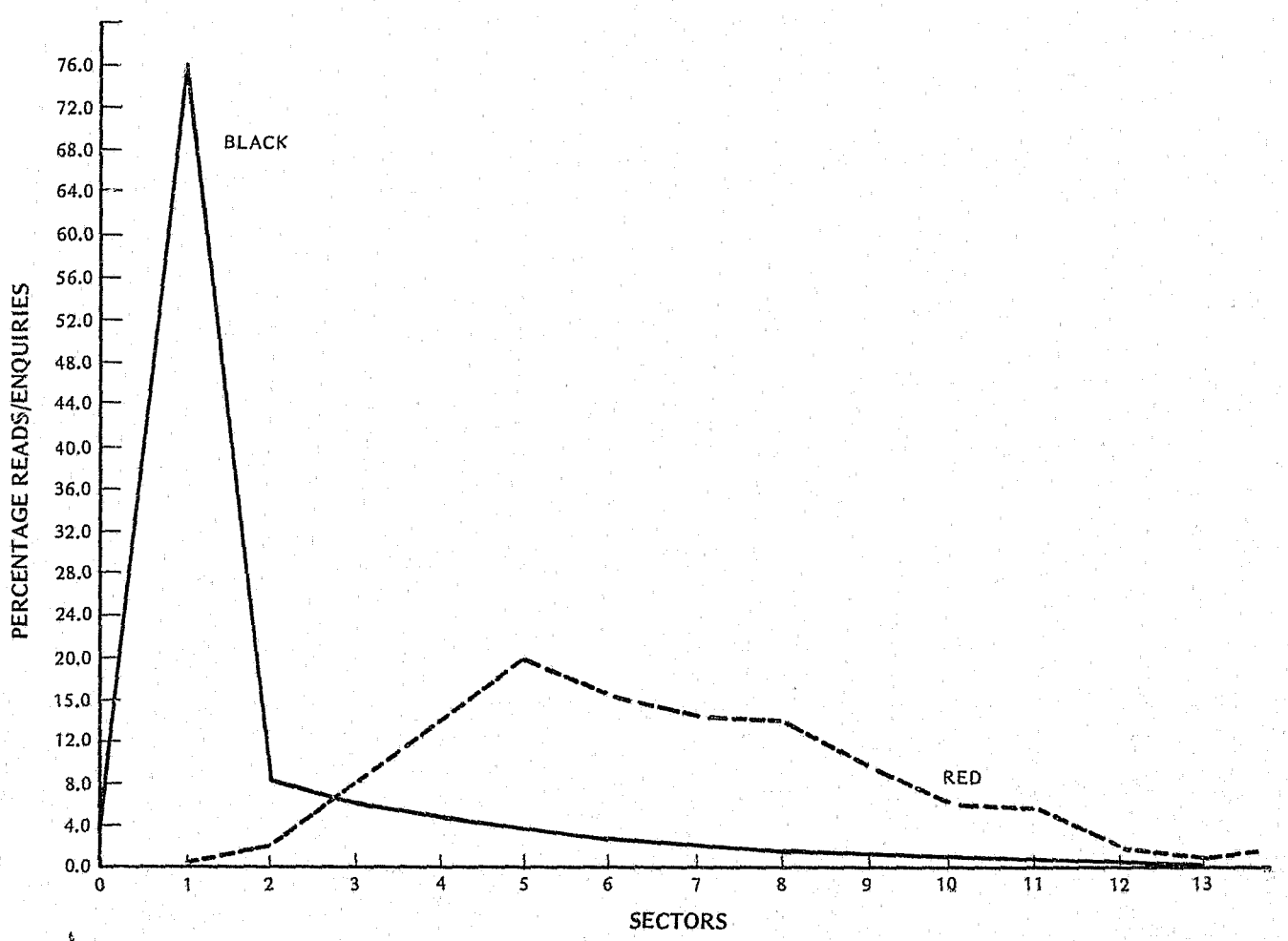
Due to the normal distribution, more than half of the transformed radix value groups use a portion of the 6th page, with an overall disk space utilization factor of 86%.

The entries contained within each radix value are maintained in descending sequence of their frequency of occurrence within the detail file. The philosophy here is that the more common the surname the more easily accessible that surname should be.

The following graph demonstrates the benefits of this sequencing. The 80% read curve (black) represents the percentage of enquiries satisfied by the number of sectors read after the sequencing of surnames. The other is the percentage of enquiries satisfied without sequencing. It can be seen that 80% of the enquiries are satisfied by a one-sector header read where sequencing has been employed, as opposed to upwards of six sectors without sequencing.

As a consequence to the employment of the previous principles, we have, in summary, been able to benefit as follows:

(1) to employ the name code as a direct access key;
(2) to save more than 25 million bytes of storage as a direct result of not repeating the surnames and, in excess of 65 million bytes by employing codes and abbreviations within the detail records. In total, 2.5 million name records have been contained within less than five disk packs of a 2314 disk drive;
(3) with the header/detail record structure effect a 98% disk utilization;
(4) by utilizing a multi-sector read facility, minimize the read time;
(5) develop a system which requires no near-future file re-organization as new records are added and others deleted.


PERCENTAGE READS/ENQUIRIES
76.0
72.0
68.0
64.0
60.0
56.0
52.0
48.0
44.0
40.0
36.0
32.0
28.0
24.0
20.0
15.0
12.0
8.0
4.0
0.0
BLACK
RED
0
1
2
3
4
5
6
7
8
9
10
11
12
13
SECTORS

END