

November 14, 2000
 Project Due: Nov. 22, 2000
 Internals → focus on notes!

Final: December 6, 2000

Pointers

Main memory (address) ONLY (as a buffer)

Dynamic vs Static

Simulate Pointers (Chaining: CPS305)

Programming Language

	<u>Pointers</u>
C/Algo	Yes
Fortran/Basic	No
COBOL	No
Ada	?
Java	?

OS: Types of Files

- Binary
- Direct Fortran
- Relative (COBOL, Data Structures)
- Access Tyeeps, etc...
- ...

BOTH IN MAIN MEMORY AND FILES

Files: (Look in old text books)

- Sequential (no use)
- Index Organized files
- List organized
- Random

Chaining: Lists (Files)

	Block	Next
1	CB	4 2
2	MAST	5
3	BB	1
4	EF	2
5	RR	7
6	WWf	0
7	WCWF	6

Rec #	Record Itself
-------	---------------

106	106	106	106
CB	MAST	BB	EF

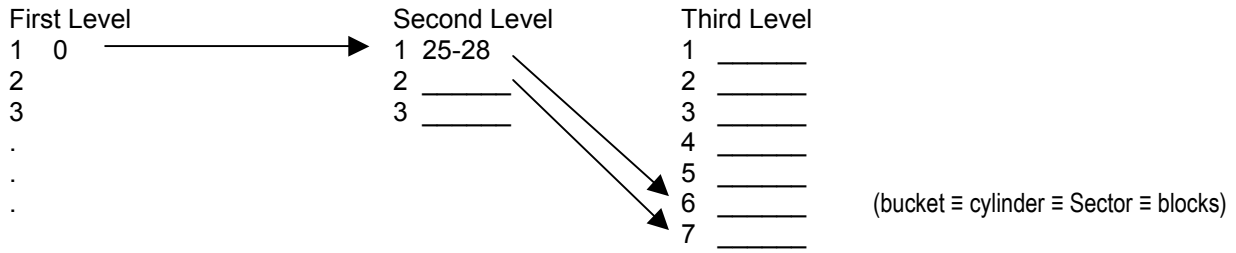
106 x 4

Start	3
END	6

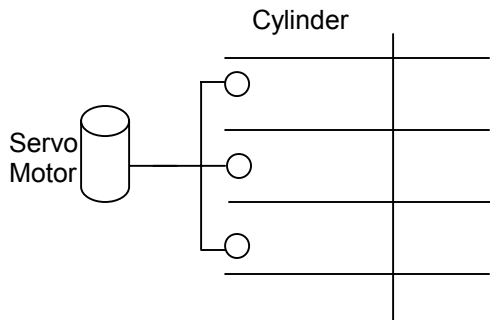
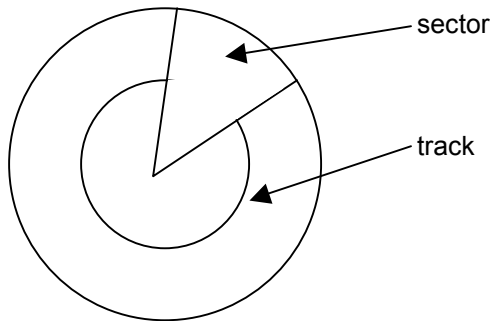
Why 3? Because system tries to put it in alphabetical order.

Delete EF
 Garbage Collect BB
 Start End CB
 4 4 EF
 MAST
 RR
 WCWF
 WWf

Hash(key): RCC \$ not the other one!!



Record
File
Medium (disk, tape)



Internals of the Paradigm

1. S

	RN	SNR	SCODE	CITY
Block 1 {	1	S1	10	Athens
	3	S2	30	Toronto
Block 2 {	2	S4	20	Kingston
	4	S5	30	Toronto

Bottom Level in sales person index

SNR	RN
S1	1
S2	3
S4	2
S5	4

Address Conversion Table

RN	Block Address
1	1
2	2
3	1
4	2

Inverted Files (List)

Bottom Level in city index

City	RN
Toronto	3, 4
Kingston	2
Athens	1

2. SP

Block	RN	SNR	PNR	QUANTITY
1	{ 1	S2	P1	200
	{ 2	S2	P3	100
2	{ 3	S4	P5	200
	{ 4	S4	P8	100
3	{ 5	S5	P1	50
	{ 6	S5	P3	50
4	{ 7	S5	P4	100
	{ 8	S5	P5	200
5	{ 9	S5	P4	100

Bottom Level Index

SNR	RN
S2	1,2
S4	3,4
S5	5,6,7,8,9

Address Conversion Tables

RN	Block Address
1	1
2	1
3	2
4	2
5	3
6	3
7	4
8	4
9	5

Search

- Search for all sales person records where city='Toronto'
- Traverse through the city index with RN's 3 & 4
 - Address conversion table and find the block numbers
- To search for all salespersons record NOT in Toronto
 - Go to city index and pickup everything else but Toronto (RNs 2 and 1)
- To search for all salespersons records with city=Toronto or city = Athens
 - Go to city index get RNs of both and merge them.
- To search for S5
 - Go to SP's index and get RN for S5 (5,6,7,8,9)

GOOD LUCK ON THE FINAL

11/21/2000

Example: The paradigm implemented with inverted files (inverted lists)

S Table

Block	Record No	SNR	SCODE	CITY
1	1	S1	10	Athen
1	3	S2	30	Toronto
2	2	S4	20	Kingston
2	4	S5	30	Toronto

Low Level
In SNR index

SNR	Record No.	Block Address
S1	1	1
S2	3	1
S4	2	2
S5	4	2

City Index

City	Record No
Toronto	3,4
Kingston	2
Athens	1

Survey of logical operator (survey)

- Implement the logical operator
 - Suppose we have p & q are search criteria that are matched by a particular Rec. No list. So, P may be an expression such as city='Toronto'
1. p AND q: The Rec. No lists for p and q are merged, so that a new Rec. No list is created with the Rec. No's belonging to both p and qs.
 2. NOT P: The complement for P's Rec. Nos: those Rec.Nos in the extraction NOT belonging to P. Eg. City <> Athens: 3,4 & 2 → merge them {2,3,4}
 3. P and (not q): same as #1 and #2 above.
 4. \exists s:p: logical expression is true, if there are records in the file S for which the expression p is True. → scan the index lists!!!
 5. \forall s:p: logical expression that is true, if there are no Rec No. in the complement for P's Rec No's.
 6. Field value >= constant:
The result is the Recs where RecNo are retrieved by using the search argument field value=constant and all the Rec No found by sequential reading from this key value on (ASC)
 7. Field Value <= constant:
Same as above except that the sequential reading of recs is the other way (DESC)

The Advantage

- tables can be re-organized without having to reorganize the indexes
- it is possible to more around (ie due to overflow in a block)
- without it being necessary to update the various indexes at the same time.
- It is enough to change just only the address of the record in the address conversion table
- Rec Nos are used to select records satisfying a logical expression

Physical

- sequential organization: individual records are dealt with according to a fixed sequence.
- Random organization: a record can be traced by transforming the identifier (key) (number or char) of the record to block address → hashing
- Index organization: records are searched for as an index that contains references to the records of the file.
- List organization: various forms which related records are chained together by pointer fields.

Types of File Organization

Characteristics	Organization			
	Sequential	Random	Index	List
Is it possible to gain direct access to records?	NO	YES	YES	NO
Is sequential treatment possible?	YES	NO	YES	YES
Can the 1:n relationship be implemented	YES	YES	YES	YES
Can the m:n relationship be implemented	NO	NO (YES)*	NO (YES)*	YES
Ordering relationship	YES	NO	YES	YES
Tree structure be implemented?	YES	NO	YES	YES
Network (graph) relationship?	NO	NO	??	YES
Order n relationship	NO	NO	??	YES

(*Yes, if n = 1 or n = 2 ... if 2, then two 1's)

Why we implemented inverted list, inverted files?

END OF CPS510 – GOOD LUCK ON FINAL

Small Note regarding project

Final Exam: Dec 6 (covers from start to end)

Project: folder with full documentation (all 3 parts)

- diskette with all files (data/source code)
- in the beginning sheet with detail instruction of how do you load and execute your program from that diskette.
- Create and drop table at will

Disclaimer:

The correctness of these notes is not to be questioned. LS is not held responsible for any errors or misleading information found in this set of notes. Unauthorized use will ruin this for the others