Exercise Session 2

Page Addressing, Update Strategies and Shadow Paging

Note: For simplicity, we will assume in exercises 1 and 2 below that database page size is equal to disk block size.

2.1 Segments and Pages, Addressing and Update Strategies

a) Explain the role of **segments** and **pages** in a DBMS

b) Explain the differences between:

- Direct and Indirect Page Addressing
- Direct and Indirect Update Strategy

2.2 Shadow Paging

a) Describe the Shadow Paging Method. What data structures are used and what happens when pages of a segment are updated?

b) Which data structures need to be written atomically to disk when a segment is closed? Why?

c) Given is a segment S1 with 8 pages, stored in a 16 block file F.

Page	1	2	3	4	5	6	7	8
Block	2	3	4	0	0	0	0	0

Lets assume that the first 8 blocks of F are used (blocks 1 thru 8), while the rest are free. Go step-by-step through the execution of the following operations on S1 and show how the auxiliary data structures (CMAP, MAP0, MAP1, V10, V11) are used:

- OpenSegment(S1)
- Read(Page 1)
- Update(Page 1)

- Read(Page 2)
- Update(Page 2)
- Update(Page 1)
- CloseSegment(S1) // Savepoint / Checkpoint

2.3   Costs of Shadow Paging

Given is a segment S of size 128 MB stored in a file F:

- Database page size = 4 KB
- Disk block size = 512 Byte
- Size of F = 2 (Size of S) = 256 MB

a) Assuming that S is full and that there are no other segments stored in F, what is the maximum number of pages that can be updated with a single opening/closing of S?

b) Calculate the sizes (in blocks) of the free space bitmap MAP0 (assuming that 1 bit is used per page) and the page table V of S (assuming that addressing is performed at the page level).

c) Calculate the number of block accesses at save/checkpoint if:

- only 1 page was updated
- all pages were updated

Assume that master record size is 2 blocks and that all updated pages need to be flushed at savepoint - i.e. none of them has already been flushed.

d) What is the overhead in c)?

2.4   Shadow Paging and Transactions

a) To what extent can the Shadow Paging Algorithm be used as a basis for implementing a Recovery Mechanism in a Transaction Processing System? What problems need to be addressed when concurrent transactions update the same segment?

b) What are the advantages/disadvantages of recovery systems based on shadowing in comparison to conventional recovery systems based on in-place updating?