- **Ex. 1**
  1. As the "referenced" in page 1 and 2 are 1, they will all be set to 0. The new one should be like

Page	Timestamp	Present	Referenced	Midified
0	6	1	0	1
1	9	1	0	0
2	9	1	0	1
3	7	1	0	0
4	4	0	0	0

  2. The table should be like

Page	Timestamp	Present	Referenced	Midified
0	6	1	0	1
1	9	1	0	0
2	9	1	0	1
3	x	x	x	x
4	4	0	0	0

- **EX. 2**
  1. a. /include/minix/callnr.h
  b. servers/pm/table.c
  c. servers/pm/proto.h
  d. servers/pm/signal.c

  2. Their order is not certain.

- **Ex. 3**
  The ext2 or second extended file system is a file system for the Linux kernel. It was initially designed by Remy Card as a replacement for the extended file system (ext). Having been designed according to the same principles as the Berkeley Fast File System from BSD, it was the first commercial-grade filesystem for Linux.

  The canonical implementation of ext2 is the "ext2fs" filesystem driver in the Linux kernel. Other implementations (of varying quality and completeness) exist in GNU Hurd, MINIX 3, some BSD kernels, in MiNT, and as third-party Microsoft Windows and macOS drivers.

  ext2 was the default filesystem in several Linux distributions, including Debian and Red Hat Linux, until supplanted more recently by ext3, which is almost completely compatible with ext2 and is a journaling file system. ext2 is still the filesystem of choice for flash-based storage media (such as SD cards and USB flash drives) because its lack of a journal increases performance and minimizes the number of writes, and flash devices have a limited number of write cycles. However, recent Linux kernels support a journal-less mode of ext4 which provides benefits not found with ext2.

The space in ext2 is split up into blocks. These blocks are grouped into block groups, analogous to cylinder groups in the Unix File System. There are typically thousands of blocks on a large file system. Data for any given file is typically contained within a single block group where possible. This is done to minimize the number of disk seeks when reading large amounts of contiguous data.

Each block group contains a copy of the superblock and block group descriptor table, and all block groups contain a block bitmap, an inode bitmap, an inode table, and finally the actual data blocks.

The superblock contains important information that is crucial to the booting of the operating system. Thus backup copies are made in multiple block groups in the file system. However, typically only the first copy of it, which is found at the first block of the file system, is used in the booting.

The group descriptor stores the location of the block bitmap, inode bitmap, and the start of the inode table for every block group. These, in turn, are stored in a group descriptor table.

Source: `https://en.wikipedia.org/wiki/Ext2`

- **Ex. 4**

  1. It is in fact possible because techniques like spin locks can help prevent the two process to access the page at the same time.

  2. For the first scenario, it is not possible since the total number of pages that should be used is 17, and $17 \times 4096 > 65536$, but for the second scenario it is possible because now there are $128 = 65536/512$ pages need to be used.

  3. No as the two are actually binded so that both of them can be found in one-level search.