Increasing the block size on a UNIX file system involves:

1. The decrease in internal fragmentation.
2. The increase in external fragmentation.
3. Decreasing the maximum possible file size.
4. Downsizing map of free blocks.

Answer:

Increasing the block size increases internal fragmentation because each block to be bigger, harder it is completely fill it with information. Decreases the external fragmentation, since there are fewer gaps between allocated blocks. The maximum file size increases, since this is obtained by multiplying the block size for the maximum number of blocks of a file. The block map size decreases as the number of blocks decreases (when these larger).

4

The special entry for directory . is:

1. An entry for the directory itself.
2. An entry for the parent directory.
3. An entry for the root directory.
4. There is no such entry.

Answer:

. is for the directory itself. .. for the parent directory. The root is denoted by /.

1

The Link order (link)

1. In the current directory, create an entry for a new subdirectory or file.
2. Allows a file or subdirectory appears in multiple directories.
3. Establishes the connection between multiple files.
4. Create a link between the files you want to belong to the same directory.

Answer:

The link command allows you to create hard links, in which multiple files revere the same data block.

2

Search time corresponds to:

1. The time taken for the replacement algorithm to select a page when a page fault occurs.
2. The time it takes to transfer data on a disk.
3. The average time it takes for the sector to be under the read / write disk.
4. The time required for the disk heads to move to the appropriate cylinder.

Answer:

Operating Systems, an applied vision. Cap. 1 page 24 \ r \ nThe search time is determined by the time required to position the arm in the desired track.

4

"The special entry for each directory "".."" (dot-dot), in systems with hierarchical directories represents:"

1. An entry for the directory itself (with a pointer to itself).
2. An entry for the parent directory (the one above it in the hierarchy).
3. An entry for the child folder (the one below in the hierarchy).
4. This entry is only available in single level directories to simulate different levels.

Answer:

Dot. for the directory itself. Dot-dot .. for the parent directory. For a child directory entry is denoted by the directory name.

2

In Unix, the inode (index node) corresponds to a board

1. Each process has associated with all the information you need and the resources it uses.
2. Each process has associated with all the information needed by the system to control its execution.
3. Each file is associated with the attributes and addresses of the blocks of the file.
4. With the working set or set of pages used in a given time by the process.

Answer:

Inodes are used in file systems to store the metadata of a file: name, access times and modification, different attributes such as permissions and addresses of actual data blocks. It is an input file (not process).

3

A user with uid and gid = 12 = 1 is the owner of a file protection mode rwxr-x ---. Another user with uid and gid = 3 = 1 wants to run the file. Can you do it?

1. Every Time.
2. Do Not Ever.
3. Only if he puts the setuid bit.
4. Only if he puts the bit getuid.

Answer:

The example user (uid = 3 and gid = 1) has permissions rx (read and execute) on the file represented in the second block of permissions so you can run it.

1

In UNIX, the user 2 puts a physical link to a file that owns the user 1. Then user 1 deletes the file. What happens when user 2 tries to open the file with the link name and taking the appropriate permissions?

1. You can not open it, since it has been deleted.
2. You can open it if before it masquerades as user 1
3.
4. You can not open it because although it has not been deleted, it is locked.

Answer:

The operation will be correct because the file is not deleted while you have a physical link.

When the physical link counter physical links of the file will go from 1 to 2. \ r \ nAl perform scavenging counter physical links will file 2 to 1. A file is not deleted until its counter physical links is less than 0 then the file was not deleted, only its counter is decreased physical links.

4

What is false about naming files in UNIX?

1. The name service is integrated with the service files.
2. Two logical names can refer to the same file.
3. The naming certain files required to specify the device that contains them.

4. We can create a name in the call to the LINK system.

Answer:

In UNIX the named file is device independent, not like in windows where you need to specify the drive (C :, D :, ...) or other systems where you have to specify the device, everything hangs from the root directory and therefore indicating that the choice of certain files named requires specifying the device is false.

3

What is WRONG on the system call link?

1. It is equivalent to copying the file.
2. It can not be done from one device to another.
3. Its effect is undone with UNLINK call.
4. Modifies some field of the inode.

Answer:

In physical link count increases physical links of the inode, melts to unlink and you can not do it from one device to another (something that if they allow symbolic links). It is not equivalent to copying a file because the copy indicates you have two different files on disk that could change independently while the link made to link provides access to a single file that has 2 different names.

1

What information does not appear in the inode of UNIX?

1. The ID of the file owner.
2. The pointer to the current position read-write.
3. The creation date of the file.
4. Rwx bits.

Answer:

The pointer to the location of access (read-write) the file is not saved in the file system but an operating system structure associated with the processes that are using the file.

2

Which of the following information is NOT on the inode?

1. File access permissions.
2. Filename.
3. Number of links to the file.
4. Group ID of the owner.

Answer:

The file name is not on the node. The directories are composed of tuples of the type (name, inode). The file system defines a directory tree, where each node includes entries for that level.

2

What is true regarding the Unix system cache files?

1. The blocks inode not kept in cache.
2. The directory blocks are not kept in cache.
3. Accelerates but scripture readings.

4. Increases performance and reduces system reliability.

Answer:

Improves performance because the blocks are written to disk rather then block copies and read if present there. The problem is that blocks are stored in the cache temporarily and are turning to block every certain period of time. If there is a memory failure or electric power between dump and dump (flush, sync) updates can be lost between the two.

4

Assuming you have in memory only the superblock, the directories are small (fits in a block) and involved the entire directory is in the same volume or partition, say how many disk accesses are needed in the following operation: open (/ etc / src / foo / ajedrez.c, O_RDONLY)

1. 5
2. 7
3. 9
4. 10

Answer:

It takes 9 accesses. For each pair / name2 are needed, one for reading block directory and find the name on the block with its corresponding inode and one to read the inode. Then, as there are 4 levels have 8 hits. We lack the final access to read the first block of file data.

3

A UNIX directory entry contains ...

1. An inode only.
2. A number of node only.
3. An inode and file name.
4. A number of inode and file name.

Answer:

UNIX directories are a list of pairs (name, number of inode). They have nothing else. They allow us to link the name of a file / directory with inode. But do not contain the inode itself. This should be sought from the number and read later \ r \ n

4

In a file system with single indexing. The first-level index block:

1. Aim to other inodes.
2. Pointing to data blocks.
3. It aims to block pointers to data blocks.
4. Points to a file.

Answer:

In a filesystem direct addressing, data blocks will be targeted directly from the inode. If there is a simple indexing, there will be at least a data block whose contents are pointers to data blocks occupied by the file. A double indexing will be at least one block that contains pointers to other blocks that contain pointers to the actual data blocks occupied by the file. And so on. \ R \ nFor Therefore, the block index class in a file system with indexing siemple contain pointers to the data blocks that form part of the file. Or inodes pointers, pointers or blocks, or a file.

2

To manage the disk space the following method:

1. It is managed through operating system calls.
2. The rule of fifty percent.
3. The bitmap.
4. Map of words.

Answer:

There are different methods to manage which blocks are occupied and which are free in a filesystem. But one of the best is that the file system containing a bitmap in which each block is represented existing file system as a bit in one array (map). This bit is active or not depending on whether the block is occupied or not. The management preferred not done by system calls, that would imply an overhead. No sense makes a map of words when only one bit is required to determine whether the block is occupied or not. The 50% rule has no bearing on this question.

3

The bitmap used:

1. To keep a track of free disk space.
2. To maintain a list of the blocks that have changed and must be updated on disk.
3. To indicate which blocks up the disk cache.
4. As a token of the signals generated periodically by the real time clock, RTR.

Answer:

A bitmap is a bit array in which each element (each bit) represents the state of an object associated with it. If the element value is 1 indicate a state of the associated object and if 0 other. In the case of a file system, you can use bitmaps to know the status of different elements, but one of the most common uses is to know the status of each existing block on the disk. If the bit is active or not, indicate whether the block is occupied or not. \ R \ nFor Therefore, the correct answer is used to keep track of free disk space. \ R \ nThe blocks in cache is typically represented as a set of blocks which are those that are cached. And one bit per block within that portfolio for (modified) are dirty or not used. The counter clock signal is usually a numerical value that is updated (+1) on each interrupt generated by the clock.

1

The bitmap to maintain free disk space will occupy:

1. Many bits as blocks have the disc.
2. Many bits as free blocks have the disc.
3. Having as many bits as disk blocks multiplied by the number of records having each block.
4. The bitmap is only used for the management of main memory.

Answer:

A bitmap is a bit array in which each element (each bit) represents the state of an object associated with it. If the bit is 1, it indicates that the associated object has a particular status and, if it is 0, it indicates that your state is another. In the file system, each entry in the bitmap represents whether the block is occupied or not. Therefore, one bit per disk block, so if the bit is a 0 or 1 indicates whether that particular block is occupied or not necessary.

1

The Link order (link)

1. In the current directory, create an entry for a new subdirectory.
2. Allows a file or subdirectory appears in multiple directories.
3. Establishes the connection between multiple files.
4. Create a link between the files you want to belong to the same directory.

Answer:

The links (symbolic or hard) they do point to a real file that is in another file system path. Therefore, it is true that allows a file or subdirectory appears in several directories (routes) different. \ R \ nNo creates new directories, but aims to existing files or directories. \ R \ nNo establishes the connection between multiple files, but with one. \ r \ nNo creates a link so multiple files belonging to the same directory (not your purpose).

2

In the method of contiguous disk space allocation:

1. The address of the first block and file length is needed.
2. A few bytes to the start of the blocks is used as a pointer to the next block, the rest contains data.
3. You need to place the block indexes files on an index table.
4. There is no such allocation method.

Answer:

" ""In contiguous allocation when the file is created is assigned a fixed number of consecutive blocks that are on the disk Therefore, the only information needed is where the file begins and the space taken \ r \ nThe event \.. """" A few bytes of the beginning of the blocks are used as a pointer to the next block, the rest contains data. \ """" explains the operation of the linked or chained assignment. \ r \ nThe indices are only required for more complex systems of allocation . """

1

The method of linked lists for allocating disk space has the following drawback:

1. You must know the maximum file size at the time of its creation.
2. The resulting external fragmentation on the disk.
3. The random access a file is extremely slow.
4. The loss of space due to index tables.

Answer:

" ""The correct answer is: \"" ""The random file access is extremely slow \"" ""because, to access any file block must always begin with the first and to jump to the next until the searched. \ r \ nThe external fragmentation is not a problem, because any block can be reassigned to another file, not being unusable between files blocks. \ r \ nIt is necessary to know the size of the previous file, since you can grow only reserving a new block and pointing his address as following the last block (this pointer is saved in the hitherto last data block). \ r \ nFor Finally, this system has no index, so it does not needs space for index tables. """

3

What kind of fragmentation presents the method of allocating contiguous space in a file on a disk?

1. Only internal fragmentation
2. Only external fragmentation
3. Internal and external fragmentation
4. No amount of fragmentation

Answer:

In allocating contiguous blocks you prevent small blocks remaining to delete files (external fragmentation) to take advantage, besides the files have any size that is not a multiple of the block size so you also have internal fragmentation.

3

Indicate which of the following statements about the cache block Filesystem is false.

1. Increases the performance of SF.
2. Normally the blocks are arranged in LRU.
3. Fits any type of device.
4. A block can be cached for over an hour.

If the block device to read or write are very large can not be profitable to have a cache block for that device.

3

Answer:

For a UNIX system with a single volume or partition, determine the minimum number of disk accesses that are necessary to perform the open operation (/etc/tpm/khju/tt/tuip.c, O_RDWR). Consider that the operation mounted loaded into memory superblock, but you do not have any information on the cache block.

1. 5
2. 10
3. 11
4. 13

Answer:

Inode access 1- / 2- data / access, 3- inode / access etc, 4-access data / etc \ r \ -N5- inode access / etc / tpm, 6- access / etc / tpm \ r \ data N7- inode / etc / tpm / khju, 8- access / etc / tpm / khju \ r \ n9- inode access / etc / tpm / khju / tt, 10- access data / etc / tpm / khju / tt \ r Data \ access n11-inode access /etc/tpm/khju/tt/tuip.c

3

In a legacy UNIX file sharing ...

1. Through the same process table entry.
2. Through the entrance of the FILP table.
3. Through the entrance of the inode table.
4. You can not share.

Answer:

Each file has its process table. The son inherits the father but have your copy. Now the files opened by the father and inherited by the child point to one input of FILP table including pairs (described, pointer position). Thus if the parent or the child change the pointer position, both see it.

2

Which of the following statements about links to files is correct?

1. The space occupied by a symbolic link is proportional to the size of the filename it points to. The occupied by a no no symbolic link.

2. The space occupied by a link, whether symbolic or not, is proportional to the size of the filename it points to.
3. The space occupied by a link, whether symbolic or not, depends on the size of the filename it points to.
4. The space occupied by a non-symbolic link is proportional to the size of the filename it points to. The occupied by a symbolic link no.

Answer:

The symbolic link is assumed that in the name and node i of a file points to another whose link is indicated by the destination file name. Therefore, the space occupied by a symbolic link is proportional to the size of the filename it points to. The occupied by a no no symbolic link.

1

Consider the call to execv system (/ dir / bin / mandate, argv) and indicate which of these reasons can cause it to fail.

1. Do not have execute permission on the / dir
2. Do not have read permission on the / dir / bin
3. Not be located in the / dir / bin
4. Not having the / dir / bin in the PATH

Answer:

To execute a command must be able to traverse directories that allow to reach him. UNIX and LINUX This implies that the / dir must have execute permissions. Otherwise it may fail.

1

Which of the following structures managing free space combined with management which blocks correspond to what files?

1. Directory CP / M.
2. DOS FAT.
3. The UNIX inodes.
4. UNIX directories.

Answer:

The DOS FAT is based on linked lists block devices. Therefore, the list indicated in device space and create a file and give blocks have to manage that list. Furthermore, it linking blocks qu_bloques known list correspond to which files. The UNIX inode is a tree structure, but have no relation to the management of free space.

2

A user owner of a file with uid=12 and gid=1 has set the following attributed: rwxr-x---. Another user with uid=3 and gid=1 wants to execute it. It is possible?

1. Only if its uid is also 12
2. Always
3. Never
4. None user (including the owner) can execute the file

Answer:

2

Which of the following information is not stored in an i-node

1. Owner ID
2. Pointer to the current file read position
3. The creation date
4. The RWX bits

Answer:

2

What does the symbol "." refer to in the context of directories

1. The root directory
2. The number of links
3. It is void (it refers to nothing)
4. The current directory

Answer:

4

A path that included the root directory is called…

1. Active
2. Passive
3. Relative
4. Absolute

Answer:

4

In C programming language when an open system call success it returns…

1. Always the value 1
2. A file descriptor
3. Nothing
4. A pointer to the file

Answer:

2

A symbolic link…

1. Cannot be deleted
2. Doesn't contain a pointer to an i-node
3. It is a file copy
4. Contain a pointer to an i-node

Answer:

2