Xi Liu, xl3504

1.
No, it is not fundamentally necessary to store on disk information about the unallocated disk sectors. Information about whether inodes or data blocks are free or allocated can be calculated by accessing the slot in the multi-level index structure. File system consistency check utilities are made easier by using allocation structures.

2.
Operations must be done to 2 replicas instead of 1, so performance becomes slower for file open, allocation, truncation, and modification operations. The process of reading and writing can be faster since one of the replicas is closer to the disk head, and that one can be chosen to be read and write instead of the further replica.

3.1.
To allocate an inode for file bar and allocate space within directory foo: traverse the inode bitmap to read to find a free inode, then write to the inode bitmap to mark the free inode allocated, write to the inode to initialize its fields, write to the directory entry structure to associate the human friendly file name with the file's inode number.

3.2.
write to inode bitmap, write to foo data, write to bar inode, write to foo inode.

3.3
The file system checker or fsck does following actions:

a. Does sanity checks on superblock, find a suspect corrupt superblock
b. Scans the multi-level index structure to see which blocks are allocated for file system, uses this information of allocation to obtain a correct version of bitmap
c. Checks that allocated inode's fields (such as permission, access time, block pointers) are valid
d. Traverses the entire directory tree to calculate the number of directories that contain a link to this file to verify link count
e. Checks duplicate pointers
f. Checks block pointers to see if they are outside their valid range
g. Checks the contents of directory such as ensuring that "." And ".." are first entries

The actions ensure file system consistency because it scan the entire disk to obtain information involved in the update of the block.

3.4
Series of writes: TxB (transaction begin), I[v2] (inode), B[v2] (bitmap), Db (data block), and TxE (transaction end).

After a crash:
If crash occurs before transaction is written to the log, then the update would be ignored
If crash occurs after the write to the transaction commit block but before the checkpoint of writing the contents of the update to the final on-disk locations, then the system boots and attempts to replay (sequentially) the transactions that have committed to the disk.

4.
Set the contents of the pointer that points to the data blocks to be zero, free the metadata blocks, set the corresponding bitmap to be free.

5.
In the case of block reuse, after the deletion of an entry in a directory and the deletion of the directory, then the user created a new file that started to reuse the same block, but the newly written file is not journaled, after a crash happens, the replay process of the redo procedure overwrites of the user data of the new file with the contents of the old directory.

6.
NFS here stands for Network File System. A remote Network File System can be slower than a local Unix file system when the network connection is unstable, or if the physical distance between the server and the client is sufficiently big. System calls or operations for one client can be affected when other clients modify the file concurrently on other machines.

7.1
True, a perfect program would not be vulnerable to buffer overflow.

7.2
False, making the stack non-executable only mitigates the problem, but arbitrary code sequences can still be executed using a return-oriented programming in which the return address points to a malicious instruction.

7.3
False, making program text read-only cannot fully solve the security problem, because the program can still obtain input from some file, the malicious input can be read into the memory of the attacked program.

7.4
False, write xor execute security policy uses a NX (no execute) bit to prevent execution for the page table entries that has this bit set. Some languages such as Perl, Python, and Java do not need write xor execute policy whereas many C programs do, the policy places needless restriction on the languages that don’t need the security policy.

7.5
False, address space layout randomization randomizes the placement of heap, stack, and code within the virtual address space, but ASLR does not provide security for the vulnerable server since server informs the client about the location of buffer. ASLR can be broken by “stack reading” a return address.

7.6
False, buffer overflow vulnerabilities are still possible on 64-bit systems, the server can re-fork children instead of restarting.

7.7
False, %cr3 control register stores the physical address of the current page table’s base address, but an attack can exploit other vulnerabilities such as unbounded copy in strcpy with an input longer that the buffer size.