Link: <https://stackoverflow.com/questions/57730592/error-reading-disk-when-disk-is-not-a-hard-drive-int-0x13-ah-0x02>

Your code reads 31 sectors starting from from CHS(0,0,2) when it loads the kernel. You don't show your kernel (kernel.bin) but I suspect that it's less than 31 sectors in size.

When you do:

qemu-system-x86\_64 -drive file=os.bin,if=floppy,index=0,media=disk,format=raw

you boot as the first floppy disk. Since QEMU generally allows you to read past the end of a floppy disk image the [Int 13h/AH=2](http://www.ctyme.com/intr/rb-0607.htm) succeeds.

When you do:

qemu-system-x86\_64 -drive file=os.bin,format=raw

you boot as the first hard disk. QEMU is likely complaining because you have requested to read 31 sectors worth of data but there isn't that much data in the disk image os.bin. I believe the general rule is that for QEMU's hard disk read to work there has to be at least 1 byte of data in a sector for the read to succeed. That would mean that at a minimum you'd have to have an os.bin that is **at least** 512 bytes (bootsector) + 30 \* 512 bytes (kernel) + 1 (at least 1 byte in the 31st sector) = 15873 bytes in size. I would expect then that if your image file is less than 15873 bytes, reading 31 sectors from CHS(0,0,2)/LBA(Logical Block Address)=1 will fail. That is likely why you are getting the error:

unsupported track or invalid media

The fix is rather simple. Make sure your os.bin is at least 32 sectors (boot sector + maximum of 31 sectors for the kernel) or a file size of 32\*512=16384. You can use the DD program to build a 16384 byte image and then use DD to place the boot.bin and kernel.bin files inside of it.

Your Makefile entry for building os.bin could probably look like (save it as “makefile):

os.bin : build/boot.bin build/kernal.bin

dd if=/dev/zero of=$@ bs=512 count=32

dd if=build/boot.bin of=$@ bs=512 conv=notrunc

dd if=build/kernal.bin of=$@ bs=512 seek=1 conv=notrunc

The first command creates a zero filled file called os.bin using a block size (bs) of 512 and generating a file with 32 blocks. 32 \* 512 = 16384. The second command writes boot.bin to the beginning of the file to block 0 (first block). The conv=notrunc says that after writing boot.bin to os.bin that we don't want the file to be truncated. The last line is similar but it writes kernal.bin to os.bin but tells DD to seek to block 1 on disk and write the file and not to truncate os.bin when finished.

After this Makefile recipe is complete you should have an os.bin file that is 16384 bytes long containing your bootloader and kernel. This should keep QEMU happy whether it is reading as a floppy or hard disk image when using Int 13h/AH=2.