Name
CprE 308
Lab 8
Date

Submit this document completed, a Makefile, and the supporting c files & image.

3.1

	Hex	ASCII
Bytes/Section	0x2250	512
Sectors/Cluster	0x22b0	16
Root Directory Entries	0x2300	224
Sectors/FAT	0x22c0	1

3.2

Name	mkdosfs
Bytes/Sector	512
Sectors/Cluster	16
Reserved Sectors	1
Number of FATS	2
Root Directory Entries	224
Logical Sectors	2880
Medium Descriptor	0x00f0
Sectors/FAT	1
Sectors/Track	18
Number of Heads	2
Number of Hidden Sectors	0

//TODO insert screenshot of code

```c
// combines 2 bytes into an integer value
int combineBytes(unsigned char buffer[], int i)
{
        return ((buffer[i+1] << 8) & 0xFF00) | (buffer[i] & 0xFF);
}


// Fills out the BootSector Struct from the buffer
void decodeBootSector(struct BootSector * pBootS, unsigned char buffer[])
{
        int i = 3;  // Skip the first 3 bytes

        // Pull the name and put it in the struct (remember to null-terminate) -- 3-10
        char* osName = calloc(9, sizeof(char));
        for(; i < 11; i++)
                osName[i-3] = buffer[i];
        osName[7] = '\0';
        strcpy(pBootS->sName, osName);
        free(osName);

        // Read bytes/sector and convert to big endian -- 11-12
        int bytesec = combineBytes(buffer, 11);
        pBootS->iBytesSector = bytesec;

        // Read sectors/cluster, Reserved sectors and Number of Fats -- 13, 14-15, 16
        int secclust = buffer[13];
        pBootS->iSectorsCluster = secclust;
        pBootS->iReservedSectors = combineBytes(buffer, 14);
        pBootS->iNumberFATs = buffer[16];

        // Read root entries, logicical sectors and medium descriptor -- 17-18, 19-20, 21
        pBootS->iRootEntries = combineBytes(buffer, 17);
        pBootS->iLogicalSectors = combineBytes(buffer, 19);

        // Use the raw hex ☺
        pBootS->xMediumDescriptor = buffer[21];

        // Read and covert sectors/fat, sectors/track, and number of heads -- 22-23, 24-25, 26-27
        pBootS->iSectorsFAT = combineBytes(buffer, 22);
        pBootS->iSectorsTrack = combineBytes(buffer, 24);
        pBootS->iHeads = combineBytes(buffer, 26);

        // Read hidden sectors -- 28-31 (4 byte value)
        int32_t hidsec = ((buffer[31]) << 24) | ((buffer[30]) << 16) | ((buffer[29]) << 8) | (buffer[28]);
        pBootS->iHiddenSectors = hidsec;

        return;
}
```