San Francisco State University

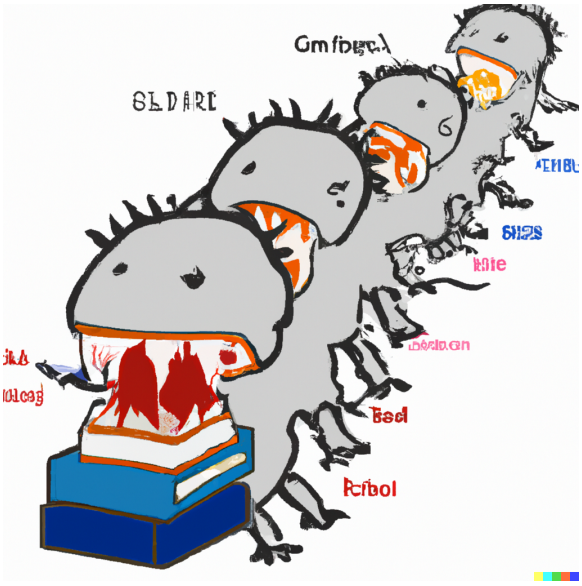
CSC415-03

File System Documentation

Milestone 1

Team Boberts
Gabriella Arcilla
Maliah Chin

Andy Cho
Leo Powers



"A Filesystem Monster made up of Contiguously Allocated Blocks"
An image created by DALL-E 2, an AI that creates a drawing based off of text input.

October 27, 2022

Github Repo:https://github.com/CSC415-2022-Fall/csc415-filesystem-boberts
Supplemental Documentation: https://boberts.surge.sh/

Contents

1 Versioning

Version	Date	Notes
0.0.0	*<2022-10-19 Wed>*	Document Initiation
0.0.1	*<2022-10-20 Thu>*	VCB and dirEntry Prototyping
0.0.2	*<2022-10-21 Fri>*	Changed from FAT -> FFS
0.0.3	*<2022-10-22 Sat>*	Fixed logic for FSM corrected syntax errors for malloc functions
0.0.4	*<2022-10-23 Sun>*	iNodes vs Extents vs Clusters Notes
0.0.5	*<2022-10-25 Tue>*	Added Team Contribution Table Added Team Methodology
0.1.0	*<2022-10-27 Thu>*	Finishing Touches for Milestone 1 and Cleanup

2 Kanban

TODO	DOING	DONE
		CODE Volume Control Block(block 0)
		CODE Free Space Management
		CODE Root Directory
		DOC Hex Dump
		DOC VCB structure
		DOC Free Space Structure
		DOC Directory System
		DOC Contribution Table
		DOC Team Methodology
		DOC Issues Faced

3 Description

3.1 Our Filesystem

Team Boberts has taken on the task of creating a filesystem for a disk drive with a contiguous array of 512 byte chunks. The freespace within our filesystem is managed through a bit vector where each block is represented by a 1 or a 0 depending on whether it is free or not. In this milestone, we were tasked to write the code to initialize our filesystem. This included performing volume checks, initializing our volume, initializing our bit vector for space management, and initializing our directory system with the root directories. Within this timeframe many issues arose, as such, we have documented them- however with the complement of hours of brainstorming and an agile state of mind, we were able to mitigate most of our issues. At this stage of the process, we need still need to come to map out how we want to manage inodes throughout our filesystem. This is our next step- to create plans, models, and implementations for the inode structure. Afterwards we need to integrate it within our filesystem, changing the pre-existing values for the VCB and FSM. Then, after fixing lower priority issues, we can move on to coding the actual command line functionality.

4 Volume Control Block Structure

The Volume Control Block Structure contains various volume details. We use inodes to directly link to the files within our file system. The variabels that contain the details of block refer to the actual space we use in the Logical Block Array. Finally, we introduce two parts of the magic numbers with variables long signatureP1 and long signatureP2, which allows us to reecognize what format our file is in.

4.1 VCB information table

ATTRIBUTE	$DATA_{TYPE}$	DESCRIPTION
label	char *	**CHANGE** This needs to be set to a const value, FAT uses 11
type	char *	
volumeName	char *	**CHANGE** This needs to be set to a const value, FAT uses 8, what does FFS use?
volumeSize	int	Given by the execution command input, default 19531
firstBlock	int	firstBlock for storing files(root)
firstFreeBlock	int	Incremented/Decremented based on CRUD operations
freeBlockCount	int	# of Freeblocks, used to calculate disk space
inodeCount	int	
inodesPerGroup	int	
freeInodeCount	int	
sizeOfRoot	int	initial size of Root
rootIndex	int	Same as firstBlock
signatureP1	long	**CHANGE** combine signature 1 and 2 and save time doing the signature check
signatureP2	long	**CHANGE** combine signature 1 and 2 and save time doing the signature check

4.2 Helper Functions

4.2.1 VCBinit1()

This is called at the beginning of fsInit and returns the first block from LBA so it can be used for signature checking.

```
VolumeControlBlock *VCBinit1() {

  VolumeControlBlock *VCB =
      (VolumeControlBlock *)malloc(sizeof(VolumeControlBlock));
  printf("\nInitializing VCB attributes...\n");

  return VCB;
}
```

4.2.2 VCBinit2()

This is called whenever the signature check fails(2 cases), and it is used to initialize the very beginning data elements needed within our superblock.

```
VolumeControlBlock *VCBinit2(VolumeControlBlock *vcb, uint64_t blockSize,
     uint64_t blockCount) {
  printf("\nInitializing VCB attributes...\n");
  vcb->signatureP1 = PART_SIGNATURE;
  vcb->signatureP2 = PART_SIGNATURE2;

  vcb->blockSize = blockSize;
  vcb->blockCount = blockCount;

  printf("\nblockSize: %lu", blockSize);
  printf("\nblockCount: %lu", blockCount);

  return vcb;
}
```

4.2.3 VCB Struct

```
/*Volume Control Block*/
typedef struct VolumeControlBlock {

  char *label;
  char *type;

  char *volumeName;
  int volumeSize;

  int firstFreeBlock;

  int inodeCount;
  int inodesPerGroup;

  int blockCount;
  int blockSize;
  int blocksPerGroup;

  int sizeOfRoot;
  int rootIndex;

  int freeBlockCount;
  int freeInodeCount;
  int firstBlock;

  long signatureP1;
  long signatureP2;

} VolumeControlBlock;
```

5 initFileSystem

5.1 Methodology

In order to initialize our filesystem, we first bring read the first block from disk using LBAread. This initial step of reading from LBA is contained in a function called VCBinit1(), and simply returns a pointer to a VCB object to be used for validity checking. We can compare the signature between the block we brought in from LBAread and that declared globally by the program. If there is a match, there is no need to initialize and format a new volume, however, by contrast, we need to take the necessary steps to mount a new volume on to disk. The logic is contained as such:

Check 1	Check 2	TODO
PASS	PASS	NO NEED TO INIT
PASS	FAIL	INIT FILE SYSTEM -> Init VCB values by calling VCBinit2(VCB, blockSize, numberOfBlocks) -> Set the firstBlock to that given by the FSM obect -> Set the rootIndex to where root starts -> LBAwrite our VCB to block 0 -> LBAwrite our FSM to block 1, up to blocksReq amount -> set sigCheck to 1
FAIL		I>NIT FILE SYSTEM -> Init VCB values by calling VCBinit2(VCB, blockSize, numberOfBlocks) -> Set the firstBlock to that given by the FSM obect -> Set the rootIndex to where root starts -> LBAwrite our VCB to block 0 -> LBAwrite our FSM to block 1, up to blocksReq amount -> set sigCheck to 1

After these conditionals, our filesystem will have been initalized(and if needed, formatted) ready to be used.

6 Free Space Management

Our implementation of Free Space Managment uses a bitmap array to organize the data. The Free Space Map contains several variables that will assist us by providing ways to track where we are within our free space map. Other data of our blocks included in our freeSpaceMap structure is the total blocks and the size of the blocks. An additional variable we use to compute the total blocks needed in the Logical Block Array to contain our free space map, we call int blockReq. We access the freeSpace map using pointers and initialize our free space map structure with parameters (int totalBlock, int blockSize), as well as allocating the proper memory to the map.

6.1 initFSP(int totalBlocks, int blockSize)

This is a function that is called when a volume needs to be initialized. The first 6 bits are marked as 0 because they are in use by the vcb and the bitmap vector. The rest are marked as free.

```
freeSpaceMap *initFSP(int totalBlocks, int blockSize) {

  freeSpaceMap *fsm = (freeSpaceMap *)malloc(sizeof(freeSpaceMap));

  fsm->blockReq = (((totalBlocks / 8) + 1) / blockSize) + 1;
  fsm->firstBlock =
      fsm->blockReq +
      1; // TODO, calculate and find the architecture for the 1st block
  fsm->lastBlock = fsm->firstBlock + fsm->blockReq;

  // fsm->map = malloc((totalBlocks / blockSize) + 1);
  printf("\ntotalBlocks/blockSize  + 1: %d", (totalBlocks / blockSize) + 1);

  // Block 0 is occupied(512, Block 1 + blockReq is occupied(blockReq * 512)
  //  [0][1][2][3][4][5][6][7][...][totalBlocks-1]
  //   v  f           f
  for (int i = 0; i < fsm->lastBlock; i++) {
    // fsm->map[i] = malloc(sizeof(int));
    fsm->map[i] = 0;
  }

  for (int i = fsm->lastBlock; i < totalBlocks; i++) {
    fsm->map[i] = 1;
  }

  // Print freeSpaceMap
  // for(int i = 0; i < totalBlocks; i++){
  //     printf("%d", fsm->map[i]);
  // }

  return fsm;
}
```

7 Directory System

The Directory System follows the steps laid out in the milestone 1 steps. First we created a pointer to a array of directory entries, then we initialize each entry to the free state. We do this inorder to track the avaliable directory entries in the directory structure. Next we called the getFreeSpace function inorder to find a continguous block chunk of the size of our directory structure. Then we wrote the blocks in to memory with LBAWrite and returned the blocks that we used inorder to save them in the volume control block.

```
typedef struct dirEntry {
  char name[255];
  char owner[25];
  unsigned int dirATTR;
  // Attribute table
  //  1: (Read-only)
  //  2:  (Hidden)
  //  4:  (System)
  //  8:  (Volume label)
  //  10:  (Directory)
  //  20:  (Archive)
  //  0F:  (LFN entry)
  unsigned int fileID;
  // use bit values to track permissions, use enum to track which permissions
  unsigned int permissions;
  unsigned int groupID;
  unsigned int inodeID;
  unsigned int pathLength;
  unsigned int startingBlock;
  unsigned int size;
  unsigned int freeFlag;
  time_t lastAccess;
  time_t lastMod;
  time_t CRTTime;
} dirEntry;
```

8 Team Methodology and Contribution Table

During this initial milestone we met online daily for a quick 10-15 minute SCRUM meet. While in the beginning we were overloaded with coursework from other entities, we were able to fastidiously optimize our efforts into this project and meet up in real life 3-4 times a week to collaboratively prototype, plan, and implement code. The contribution metrics on github classrooms do not reflect the efforts of our team members because a good chunk of our code was written using an IDE sharing application- therefore only one member had to push after a days worth of work. Each of us played a significant role in deciding how we want to implement our VCB, FSM, and directory system. We were then able to divvy up tasks efficiently throughout our team.

BOBERTS Team Member	Contributions
Gabriella Arcilla	Documentation Contributor Prototyping FSM Implementation GitHub Master
Maliah Chin	Documentation Contributor Prototyping SCRUM Master FSM Implementation
Andy Cho	Documentation Contributor Prototyping Volume Control Block FSM Implementation
Leo Powers	Documentation Contributor Prototyping '.' and '..' directories Directory Structure Architecture FileControlBlock(DirEntry) Prototyping and Implementation

9 Issues Faced

9.1 Carelessly Delving into an Implementation

Seeing how there are numerous ways to allocate memory, manage freespace, and optimize disk usage- we initially saw the FAT32 file system, and how it uses a combination of the Master File Table and clustering, to intuitevely map out files. We had our minds set out on implementing a filesystem based on the FAT specifications however we also saw that while clustering is a simple means of file organization, modern systems are skewing towards the usage of inodes and extents. We also realized that while we may have a handy trove of FAT32 documentation resources, we want to venture out and try implementing a system with inodes like the Unix File System and then later we can start thinking about how we can create a combined filesystem to better replicate the FFS.

Subsequently, since we wanted to use a bitmap and inodes, we had to rapidly reprototype the way we structure our VCB, FSM, and directory architecture. Bits and pieces need maintenence and they are cataloged within our documentation website under the Kanban section.

9.2 Memory Faults

While implementing our initial project we were met with cascades of memory faults. We realized that while editing in the terminal may be faster(especially in a vm) and conveinient, it requires a lot of configuration to create a development environment capable of debugging on the fly. So we changed the IDE in our tech stack to CLion because as students we are given a membership to the ultimate edition, making it easier to track memory faults. This, in combination with valgrind helped the debugging process a lot.

9.3 Initial Team Performance

Although we started thinking about the project early on, many of us were overwhelmed by other work, personal matters, and homework. In order to mitigate this, we changed some of our meetings to online and instead of having long multiple-hour meetings, we decided it would be better to have a quick 10-15minute SCRUM meet to discuss our daily Kanban and other project matters.

10 Current Issues

10.1 Mise en Place

We need to stray away from diving head-first into implementation and instead write all the pseudocode and logic for what we plan to do. Because we don't do this, each of us has a different idea of what the implementation may/should look like. Doing this, allows us to have a plan that is pragmatic and ubiquitous to all of us.

10.2 Diagramming

Diagramming will help us. Make some diagrams.

11 Hex Dump

Full Dump available at: `http://boberts.surge.sh/dump2`

```
Dumping file ../SampleVolume, starting at block 0 for 19532 blocks:

000000: 43 53 43 2D 34 31 35 20  2D 20 4F 70 65 72 61 74 | CSC-415 - Operat
000010: 69 6E 67 20 53 79 73 74  65 6D 73 20 46 69 6C 65 | ing Systems File
000020: 20 53 79 73 74 65 6D 20  50 61 72 74 69 74 69 6F |  System Partitio
000030: 6E 20 48 65 61 64 65 72  0A 0A 00 00 00 00 00 00 | n Header........
000040: 42 20 74 72 65 62 6F 52  00 96 98 00 00 00 00 00 | B treboR......
000050: 00 02 00 00 00 00 00 00  4B 4C 00 00 00 00 00 00 | ........KL......
000060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000070: 52 6F 62 65 72 74 20 42  55 6E 74 69 74 6C 65 64 | Robert BUntitled
000080: 0A 0A 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

000100: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000110: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000120: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000130: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000140: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000150: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000160: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000170: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000180: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

000200: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000210: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000220: 00 00 00 00 00 00 00 00  4B 4C 00 00 00 02 00 00 | ........KL......
000230: 00 00 00 00 00 00 00 00  06 00 00 00 00 00 00 00 | ................
000240: 00 00 00 00 06 00 00 00  42 20 74 72 65 62 6F 52 | ........B treboR
000250: 52 6F 62 65 72 74 20 42  51 31 01 00 00 00 00 00 | Robert BQ1......
000260: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000280: 00 00 00 00 00 00 00 00  00 00 00 00 01 00 00 00 | ................
000290: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0002A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0002B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0002C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
```

```
0002D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0002E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0002F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000300: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000310: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000320: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000330: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000340: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000350: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000360: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000370: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000380: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000390: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0003A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0003B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0003C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0003D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0003E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0003F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000400: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000410: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000420: 00 00 00 00 00 00 00 00  00 00 00 00 01 00 00 00 | ................
000430: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000440: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000450: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000460: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000470: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000480: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000490: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0004A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0004B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0004C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0004D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0004E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0004F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000500: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000510: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000520: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000530: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000540: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000550: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000560: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000570: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000580: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000590: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0005A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0005B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0005C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0005D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0005E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
```

```
0005F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000600: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000610: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000620: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000630: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000640: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000650: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000660: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000670: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000680: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000690: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0006A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0006B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0006C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0006D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0006E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0006F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000700: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000710: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000720: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000730: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000740: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000750: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000760: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000770: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000780: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000790: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0007A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0007B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0007C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0007D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0007E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0007F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000800: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000810: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000820: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000830: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000840: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000850: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000860: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000870: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000880: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000890: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0008A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0008B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0008C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0008D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0008E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0008F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
```

```
000900: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000910: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000920: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000930: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000940: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000950: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000960: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000970: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000980: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000990: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0009A0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0009B0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0009C0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0009D0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0009E0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
0009F0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000A00: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A10: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A20: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A30: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A40: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A50: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A60: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A70: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A80: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000A90: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000AA0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000AB0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000AC0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000AD0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000AE0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000AF0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000B00: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B10: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B20: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B30: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B40: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B50: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B60: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B70: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B80: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000B90: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000BA0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000BB0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000BC0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000BD0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000BE0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000BF0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000C00: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C10: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
```

```
000C20: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C30: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C40: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C50: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C60: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C70: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C80: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000C90: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000CA0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000CB0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000CC0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000CD0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000CE0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000CF0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000D00: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D10: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D20: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D30: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D40: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D50: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D60: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D70: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D80: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000D90: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000DA0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000DB0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000DC0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000DD0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000DE0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................
000DF0: 01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00 | ................

000E00: 2E 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E30: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E40: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E50: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E60: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E70: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EB0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EC0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000ED0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EE0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EF0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

000F00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F10: 00 00 00 00 00 00 00 00  0A 00 00 00 00 00 00 00 | ................
000F20: 00 00 00 00 00 00 00 00  00 00 00 00 06 00 00 00 | ................
000F30: A0 41 00 00 01 00 00 00  F5 38 5B 63 00 00 00 00 | A......8[c....
```

```
000F40: F5 38 5B 63 00 00 00 00  F5 38 5B 63 00 00 00 00 | 8[c....8[c....
000F50: 2E 2E 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```