1. A Little Program that creates a very small file system on a disc.
   The design of the file system is far too simple, this just illustrates a short-cut to getting a disc operational.

2. Pattern files based on the various block structs used by the file system.

3. Using the pattern files under the emulator to examine the emulated disc drive.

1. Disc creation utility

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <time.h>

typedef unsigned char byte;
byte block[512];
int file;

struct superblock
{ char name[32];
  int rootdir;
  int firstfreeblock;
  char padding[472]; };

struct fileheader
{ int creationdate;
  int length;
  byte pointerlevels;
  byte type;
  char comment[102];
  int pointer[100]; };

struct direntry
{ char name[28];
  int blocknum; };

struct dirblock
{ direntry entry[16]; };

struct freeblock
{ int nextfree;
  char filler[508]; };

void blank()
{ for (int i=0; i<512; i+=1)
    block[i]=0; }

void write(int bn)
{ lseek(file, bn*512, SEEK_SET);
  write(file, block, 512); }

void read(int bn)
{ lseek(file, bn*512, SEEK_SET);
  read(file, block, 512); }

void main()
{ if (sizeof(superblock)!=512)
  { printf("superblock wrong\n");
```

```
    exit(1); }
  if (sizeof(fileheader)!=512)
  { printf("fileheader wrong\n");
    exit(1); }
  if (sizeof(dirblock)!=512)
  { printf("dirblock wrong\n");
    exit(1); }
  file = open("maindrive.disc", O_WRONLY | O_TRUNC | O_CREAT | O_BINARY,
0600);
  if (file<0)
  { printf("Failed to write the file, don't know why\n");
    exit(1); }

  superblock * sb = (superblock *)block;
  fileheader * fh = (fileheader *)block;
  dirblock * db = (dirblock *)block;
  char * tb = (char *)block;
  freeblock * fb = (freeblock *)block;

  blank();
  strcpy(sb->name, "MyLittleDisc");
  sb->rootdir=1;
  sb->firstfreeblock=10;
  write(0);

  blank();
  fh->creationdate=time(NULL)-946702800-7200;
  fh->length=96;
  fh->pointerlevels=1;
  fh->type='D';
  fh->pointer[0]=2;
  write(1);

  blank();
  strcpy(db->entry[0].name, "one.txt");
  db->entry[0].blocknum=3;
  strcpy(db->entry[1].name, "two.txt");
  db->entry[1].blocknum=4;
  strcpy(db->entry[2].name, "three.txt");
  db->entry[2].blocknum=7;
  write(2);

  blank();
  fh->creationdate=time(NULL)-946702800-3600;
  fh->length=17;
  fh->pointerlevels=1;
  fh->type='F';
  fh->pointer[0]=5;
  write(3);

  blank();
  fh->creationdate=time(NULL)-946702800-3600;
  fh->length=1000;
  fh->pointerlevels=1;
```

```
fh->type='F';
fh->pointer[0]=6;
fh->pointer[1]=9;
write(4);

blank();
strcpy(tb, "This is file one\n");
write(5);

blank();
char c='a';
for (int i=0; i<512; i+=1)
{ tb[i]=c;
  c+=1;
  if (c>'z') c='a'; }
write(6);

blank();
fh->creationdate=time(NULL)-946702800-1234;
fh->length=19;
fh->pointerlevels=1;
fh->type='F';
fh->pointer[0]=8;
write(7);

blank();
strcpy(tb, "This is file three\n");
write(8);

blank();
for (int i=512; i<1000; i+=1)
{ tb[i-512]=c;
  c+=1;
  if (c>'z') c='a'; }
write(9);

blank();
for (int bn=10; bn<20; bn+=1)
{ fb->nextfree=bn+1;
  write(10); }
fb->nextfree=0;
write(20); }

close(file); }
```

2. Pattern files

Pattern files have names ending in “`.pat`”. The emulator reads them when they are referred to.

In a pattern file these letters have special meanings:

c read one byte, print as a character
b read one byte, print as a number
g read one byte and ignore it
s read two bytes, print value as a 16 bit int
i read four bytes, print value as an int
t read four bytes, print value as a time and date, assuming that it is a count of seconds since 0:00 1st January 2000, GMT.

This is the file `superblock.pat`

```
32*c"\nroot dir = "i
"\nfirst free = "i
"\n"
472*g
```

This is the file `header.pat`

```
"created "t
"\nlength "i
"\nlevels "b
"\ntype "c
102*g
"\npointers: "
100*i
```

This is the file `dirblock.pat`

```
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
28*c" at "i"\n"
```

This is the file `freeblock.pat`

```
"nextfree "i
508*g
```

3. Examining Disc Blocks under the Emulator

the emulator command “`disc n`” is used to select different emulated discs. By default disc 1 is selected. By default disc 1 is the file `maindrive.disc`.

the command “`block n p`” displays the contents of block n of the current disc according to the pattern `p`. Pre-defined pattern names are `chars`, `bytes`, and `ints`. Any other pattern name is assumed to be defined in a pattern file with the appropriate name (plus “`.pat`”)

This is a transcript of the emulated disc being inspected.

```
$ emulator
disc 1 from file 'maindrive.disc', 6000 blocks
set SP to 0x80000000
set PC to 0x00000400

> block 0 superblock
  1: MyLittleDisc\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
 32: root dir = 1
 36: first free = 10

> block 1 header
  0: created Wed Oct 20 16:54:54 2010
  4: length 96
  8: levels 1
  9: type D
112: pointers: 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
252: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
416: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

> block 2 dirblock
  1: one.txt\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 3
 33: two.txt\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 4
 65: three.txt\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 7
 97: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
129: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
161: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
193: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
225: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
257: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
289: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
321: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
353: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
385: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
417: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
449: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0
481: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 at 0

> block 3 header
  0: created Wed Oct 20 17:54:54 2010
  4: length 17
  8: levels 1
  9: type F
112: pointers: 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
252: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
416: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
> block 5 chars
  1: This is file one\n\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
 60: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
111: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
162: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
213: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
264: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
315: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
366: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
417: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
468: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0

> block 4 header
  0: created Wed Oct 20 17:54:54 2010
  4: length 1000
  8: levels 1
  9: type F
112: pointers: 6 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
252: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
416: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

> block 6 chars
  1: abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyza
102: xyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwx
203: uvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstu
304: rstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqr
405: opqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmno
506: lmnopqr

> block 9 chars
  1: stuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrs
102: pqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnop
203: mnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklm
304: jklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghij
405: ghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefg
498: \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0

> block 7 header
  0: created Wed Oct 20 18:34:20 2010
  4: length 19
  8: levels 1
  9: type F
112: pointers: 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
252: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
416: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

> block 8 chars
  1: This is file three\n\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
etc etc etc

> block 10 freeblock
  0: nextfree 11
> block 11 freeblock
  0: nextfree 12
> block 18 freeblock
  0: nextfree 19
> block 19 freeblock
  0: nextfree 0
> exit
```