

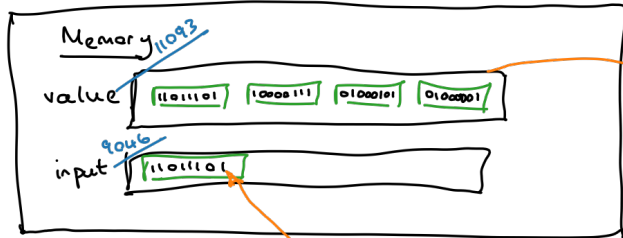
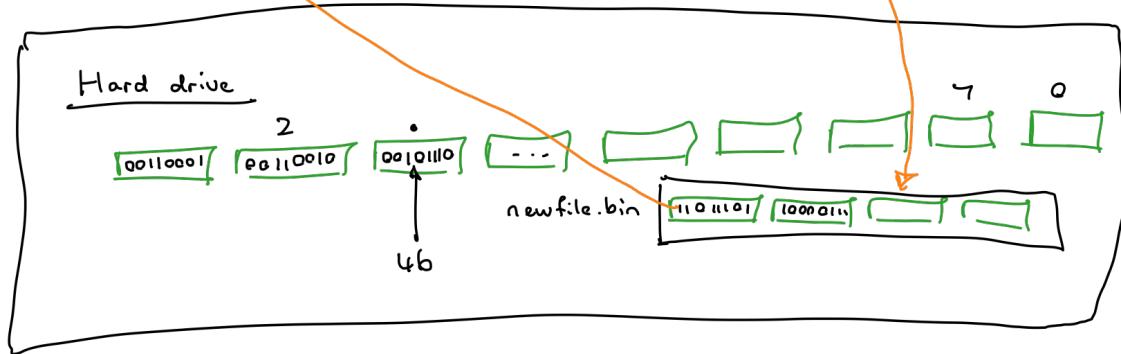
Today: Reading and writing binary files

Computer Programming 143 – Lecture 25

File Processing II

Faculty of Engineering





The bits on disk

Binary bits in value.txt:

00110001 00110010 00101110 00110011 00110100 00110101 00110110 00110111 00110000

49 46

1 2 3

ASCII table

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
46	.	57	9	75	K	86	V	103	g	114	r
47	/	65	A	76	L	87	W	104	h	115	s
48	0	66	B	77	M	88	X	105	i	116	t
49	1	67	C	78	N	89	Y	106	j	117	u
50	2	68	D	79	O	90	Z	107	k	118	v
51	3	69	E	80	P	97	a	108	l	119	w
52	4	70	F	81	Q	98	b	109	m	120	x
53	5	71	G	82	R	99	c	110	n	121	y
54	6	72	H	83	S	100	d	111	o	122	z
55	7	73	I	84	T	101	e	112	p		
56	8	74	J	85	U	102	f	113	q		

The bits in memory vs the bits on disk

```
float value = 12.34567;
```

```
11011101 10000111 01000101 01000001
```

Binary bits in value.txt:

```
00110001 00110010 00101110 00110011 00110100 00110101 00110110 00110111 00110000
```

11.6 Write to/Read from Binary File I

Unformatted I/O functions

- `fwrite`
 - Transfer bytes from a location in memory to a file
- `fread`
 - Transfer bytes from a file to a location in memory

- Example:

```
fwrite( &number, sizeof( int ), 1, fPtr );
```

- `&number` – Location to transfer bytes from
- `sizeof( int )` – Number of bytes to transfer
- `1` – For arrays, number of elements to transfer. In this case, “one element” of an array is being transferred
- `fPtr` – File to transfer to or from

11.6 Write to/Read from Binary File II

Writing structures

```
fwrite( &myObj, sizeof( struct myStrct ), 1, fPtr );
```

- sizeof – returns size in bytes of object in parentheses

Reading structures

```
fread( &myObj, sizeof( struct myStrct ), 1, fPtr );
```

- sizeof – returns size in bytes of object in parentheses

Writing and reading array of structures

```
fwrite( myArr, sizeof( struct myStrct ), numElements, fPtr );
```

```
fread( myArr, sizeof( struct myStrct ), numElements, fPtr );
```

- numElements – is the number of array elements

11.8 Sequential Read from Binary File I

```
/* Example of reading sequentially from a binary file */
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

struct coord { // structure to store coordinates
    double x;
    double y;
}; // end structure coord
typedef struct coord Coord;

int main( void ) {
    char filename[20]; // string to store filename
    FILE *cfPtr;       // file pointer
    int i;              // counter
    Coord vertices[3]; // array for triangle coordinates
```

11.8 Sequential Read from Binary File II

```
printf( "Please enter filename: " );
scanf( "%s", filename ); // reads filename from user
cfPtr = fopen( filename, "rb" ); // opens file for reading
if ( cfPtr == NULL ) { // exits if file cannot be opened
    printf( "ERROR - file could not be opened!\n" );
}
else {
    printf( "\n" );
    for ( i = 0; i < 3; i++ ) {
        // reads one set or coordinates
        fread( &vertices[i], sizeof( Coord ), 1, cfPtr );
        printf( "Vertex %d: ( %.1lf, %.1lf )\n", i + 1,
            vertices[i].x, vertices[i].y );
    }
    fclose( cfPtr ); // closes file
}
return 0; // indicates successful termination
} // end main
```

Text vs binary files

Text files

`fprintf(..)`

`fscanf(..)`

Binary files

`fwrite(..)`

`fread(..)`