CS 240 Programming in C

Typecast of Pointers, Endianess

November 13, 2019

Typecast of Pointers

Pointer can be type cast to different pointer type. But it is prone to error if you do not understand what's going on with them.

For example:

```
char s[] = "Hello";
int i = *(int *) s;

char c = *(char *) &i;
```

Endianness

1. In computing, endianness refers to the order of bytes (or sometimes bits) within a binary representation of a number.
   `https://en.wikipedia.org/wiki/Endianness`
2. Big Endian Byte Order: The most significant byte (the "big end") of the data goes first
3. Little Endian Byte Order: The least significant byte (the "little end") of the data goes first.

```
For example: short i = 0x1234;

In memory:
Big endian of byte order: [0x12,0x34]
Little endian of byte order: [0x34,0x12]
```

Endianness

1. Historically, various methods of endianness have been used in computing, including exotic forms such as middle-endianness.
2. Today, however, big-endianness of byte is the dominant ordering in networking protocols (IP, TCP, UDP).
3. little-endianness of byte is the dominant ordering for processor architectures (x86, most ARM implementations) and their associated memory.

Direct input and Output Functions

```
size_t fread(void *ptr, size_t size,
                              size_t nobj, FILE *stream)
size_t fwrite(const void *ptr, size_t size,
                              size_t nobj, FILE *stream)
```

size_t is just unsigned long;
size: data type size;
nobj: the number of this data