Common Defects in C

Alignment and packing

The device A communicates with device B by a protocol. In this protocol, each transaction conveys a formatted message.

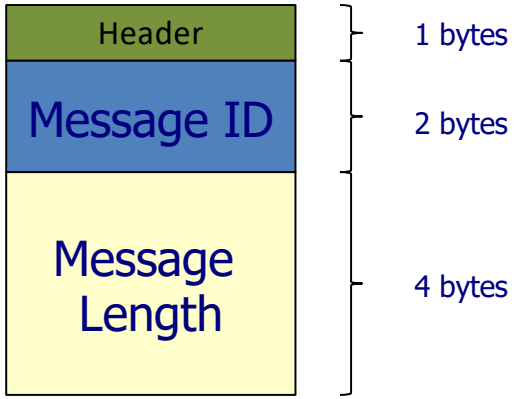
Protocol

Device A

Device B

Alignment and packing (cont) 1

- The format of message:



Alignment and packing (cont) 2

- The solution to get fields of message
  - ✓ Create a struct which fields as the message format

```
typedef struct _MESSAGE_HEADER {
     char            head_byte;
     short           message_id;
     unsigned int    length;
} MESSAGE_HEADER, * MESSAGE_HEADER_PTR;
```

  - ✓ Declare a pointer which type is above struct and cast to data received

```
void receive_data_complete( void *buffer) /* Buffer is received message data */
{
     MESSAGE_HEADER_PTR p_message_header;
     p_message_header = (MESSAGE_HEADER_PTR)buffer;
     /* Add code here */
}
```

- Are there any problems? If yes, how to fix them?

Alignment and packing (cont) 3

```
struct struct1{
    char a;
    short b;
    char c;
}
```

```
struct struct2{
    char a;
    int b;
    char c;
}
```

```
struct struct3{
    char a;
    double b;
    char c;
}
```

```
struct struct4{
    char a;
    double* b;
    char c;
}
```

```
struct struct5{
    char a;
    struct struct3
b;
    char c;
}
```

- What are the size of these structs?

(Assumption: int(4 bytes), double(8 bytes))

#define

- #define CUBE(x) x*x*x

  int x = 3;

  what will be CUBE(x + 1)?

- #define DOUBLE(x) x+x

  int x = 3;

  what will be DOUBLE((++x))?

  => How to fix it?

#define_2

- #define max(a,b) ((a) > (b)? (a): (b))

  is this always run true???

- biggest = x[0]

  int i = 1;

  while(i<n)

  biggest = max(biggest, x[i++]);

  biggest will be the biggest number in x array?

- Never pass an expression that has side effects as a macro argument

Overflow and underflow

- int x;

  Consider to x*3/5. What problem in this expression?

- Can be fix that: (x/5)*3 ?

- int x;

  (float) (3/5) * x; ?

const

- const char *p;

  char * const p;

  what is difference?

- const char *p;

  const char p[];

  what is difference?

Type

- unsigned char c;

  c = '\xff';

  if (c != '\xff') printf("Impossible!\n")

  else printf("Possible!\n")

  what it will print?

- char *p = "ab";

  char p1[2] = {‘a’, ‘b’};

  are they identical?

Operator

- if(-5 <= x <= 5){…}

  is it wrong?

  what it mean?

- if (x < 0) {

  printf("Invalid value.\n");

  exit;

  }

  is it exit if x is a negative number?

array

- int x[10][10];

  int y = x[++i, ++j];
- C doesn't actually have true multi-dimensional arrays
- x[++i, ++j] ~ *(x+(++j)) which is an address, not an integer.
- In C, always use one pair of [] for each level of array subscripting

Strings and Characters

- char c = '\n'
- char *p = "\n"

  printf("%s", &c);

  printf("%s", p);

  is it the same?

  How to fix?

Precedence

- r to an 8-bit value whose low-order bits are those of l and whose high-order bits are those of h:
  r = h<<4 + l;
- but the real mean: r = h << (4 + l);

  How to fix?
- r = (h << 4) + l;
- r = h << 4 | l;
- *p++ is ?

Precedence (cont.)

- Arithmetic operators (++, --, +, -, …)
- Shift Operators (<<, >>)
- Relation Operator (==, !=, <, <=, >, >=)
- Logical Operators (&&, ||)
- Assignment Operators

Syntax

- if (xcnt < 2)
  return;
  date = x[0];
  time = x[1];
  what it mean?
- int x = 3;
  int *p = &x;
  int y = x /* p point to x */;
  what is value of y?

Syntax (cont.)

- int *g(), (*h)();
  are these the same?
- struct foo{
  int x;
  }
  f() {
  …
  }
  what's problem?

Pointer

- char * curstr;
  char * prvstr;
  curstr = (char *) malloc(10);
  prvstr = (char *) malloc(10);
  strcpy(curstr, "abc");
  prvstr = curstr;
  strcpy(curstr, "xyz");
  what is prvstr value?
- *prvstr = *curstr; ?

Misc

- char *p;
  if(p == (char *) 0) …
  if(strcmp(p, (char *) 0) == 0) …
  are they the same?
- char c;
  while ((c = getchar()) != EOF)
    putchar(c);
  what is problem?

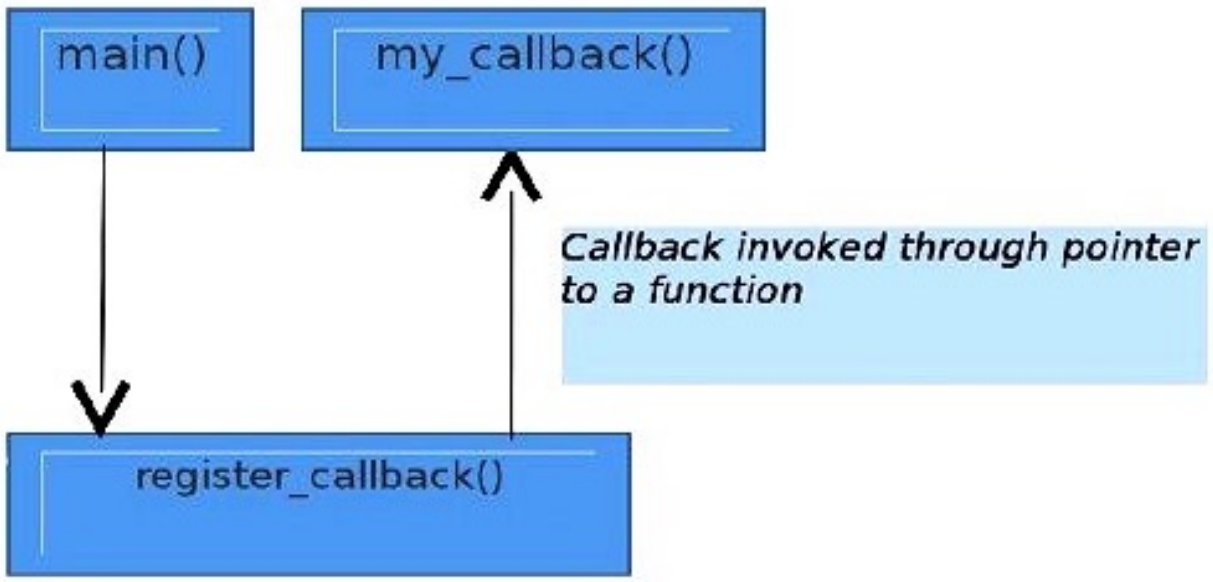
Function pointer

- Definition: Function pointer is a pointer that points to functions
- Declaration:

  <return_type> (* pfunc)(arg1, arg2);
- Purpose
  - ✓ Menu implementation
  - ✓ Callback function

Function pointer

- Callback function



Thank you

Q&A