CS356: C Programming Language Review

W. Michael Petullo

University of Wisconsin–La Crosse

As of September 8, 2021

C: For loop

```c
#include <stdio.h>

int main(void)
{
    for (int i = 0; i < 10; i++) {
        printf("%d\n", i);
    }
}
```

C: For loop

Address	Machine Code	Assembly		Meaning
401126	55	push	%rbp	Push FP
401127	48 89 e5	mov	%rsp,%rbp	Set FP
40112a	48 83 ec 10	sub	$0x10,%rsp	Grow stack
40112e	c7 45 fc 00 00 00 00	movl	$0x0,-0x4(%rbp)	Initialize i
401135	eb 18	jmp	40114f <main+0x29>	Jump to cmpl
401137	8b 45 fc	mov	-0x4(%rbp),%eax	Move i to eax
40113a	89 c6	mov	%eax,%esi	2nd printf arg.
40113c	bf 10 20 40 00	mov	$0x402010,%edi	1st printf arg.
401141	b8 00 00 00 00	mov	$0x0,%eax	Clear ret. reg.
401146	e8 e5 fe ff ff	callq	401030 <printf@plt>	Call printf
40114b	83 45 fc 01	addl	$0x1,-0x4(%rbp)	Increment i
40114f	83 7d fc 09	cmpl	$0x9,-0x4(%rbp)	Compare i to 9
401153	7e e2	jle	401137 <main+0x11>	Jump if $\leq$
401155	b8 00 00 00 00	mov	$0x0,%eax	Set return value
40115a	c9	leaveq		Restore FP
40115b	c3	retq		Return

C: An undergrad's best friend

```c
#include <stdio.h>

int main(void)
{
    printf("%lx\n", *(long *) 0x0);
}
```

C: Printf

```c
#include <stdio.h>

int main(void)
{
    const char *str = "Hello, world!";

    printf("%f %d %s\n", str, str, str);
}
```

C: Argv

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    for (int i = 0; i < argc; i++) {
        printf("%s\n", argv[i]);
    }
}
```

C: Type games

```c
#include <stdio.h>
#include <string.h>

struct foo {
    int x;
    int y;
};

int main(int c, char *a[])
{
    size_t s;
    struct foo f;

    s = sizeof f > strlen(a[0]) ? strlen(a[0]) : sizeof f;
    memcpy(&f, a[0], s);
    printf("%d %d\n", f.x, f.y);
}
```

C: Endianness

```c
#include <stdio.h>

int main(void)
{
    short n = 0x1234;
    printf("%x\n", n);
    printf("%x%x\n",
           ((unsigned char *) &n)[0],
           ((unsigned char *) &n)[1]);
}
```

C: Sizes

```c
#include <stdio.h>

int main(void)
{
    char c = 0x01;
    short s = 0x01;
    int i = 0x01;
    long l = 0x01;
    size_t z = 0x01;
    char *str = "Hello, world!";
    char array[] = "Hello, world!";

    printf("%d %d %d %d %d %d\n",
           sizeof c, sizeof s,
           sizeof i, sizeof l,
           sizeof z, sizeof str,
           sizeof array);
}
```

C: Sscanf

```c
#include <stdio.h>

int main(void)
{
    float f;
    int i, count;
    char s[BUFSIZ];

    const char *in = "3.14 42 Hello, world!";

    count = sscanf(in, "%f %d %s", &f, &i, s);

    printf("%d %f %d %s\n", count, f, i, s);
}
```

C: Pointers

```
1
2  int i = 42; printf("%d\n", i);
3
4  int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);
5
6  char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
       printf("%s\n", s);
7
8  int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
       printf("%d\n", **ipp);
9
10 void *fn();
11
12 void (*fn)(void);
13
14 int (*fn)(int);
```

C: Pointers

```c
/* Integer: */
int i = 42; printf("%d\n", i);

int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);

char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
    printf("%s\n", s);

int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
    printf("%d\n", **ipp);

void *fn();

void (*fn)(void);

int (*fn)(int);
```

C: Pointers

```c
/* Integer: */
int i = 42; printf("%d\n", i);
/* Pointer to an integer: */
int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);

char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
    printf("%s\n", s);

int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
    printf("%d\n", **ipp);

void *fn();

void (*fn)(void);

int (*fn)(int);
```

C: Pointers

```c
/* Integer: */
int i = 42; printf("%d\n", i);
/* Pointer to an integer: */
int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);
/* Pointer to a char: */
char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
    printf("%s\n", s);

int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
    printf("%d\n", **ipp);

void *fn();

void (*fn)(void);

int (*fn)(int);
```

C: Pointers

```
1  /* Integer: */
2  int i = 42; printf("%d\n", i);
3  /* Pointer to an integer: */
4  int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);
5  /* Pointer to a char: */
6  char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
       printf("%s\n", s);
7  /* Pointer to a pointer to an integer: */
8  int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
       printf("%d\n", **ipp);
9
10 void *fn();
11
12 void (*fn)(void);
13
14 int (*fn)(int);
```

C: Pointers

```c
/* Integer: */
int i = 42; printf("%d\n", i);
/* Pointer to an integer: */
int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);
/* Pointer to a char: */
char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
    printf("%s\n", s);
/* Pointer to a pointer to an integer: */
int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
    printf("%d\n", **ipp);
/* Function returning a pointer to void: */
void *fn();

void (*fn)(void);

int (*fn)(int);
```

C: Pointers

```c
/* Integer: */
int i = 42; printf("%d\n", i);
/* Pointer to an integer: */
int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);
/* Pointer to a char: */
char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
    printf("%s\n", s);
/* Pointer to a pointer to an integer: */
int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
    printf("%d\n", **ipp);
/* Function returning a pointer to void: */
void *fn();
/* Pointer to function returning void: */
void (*fn)(void);

int (*fn)(int);
```

C: Pointers

```c
/* Integer: */
int i = 42; printf("%d\n", i);
/* Pointer to an integer: */
int *ip = &i; printf("%p\n", ip); printf("%d\n", *ip);
/* Pointer to a char: */
char *s = "abc"; printf("%p\n", s); printf("%c\n", *s);
    printf("%s\n", s);
/* Pointer to a pointer to an integer: */
int **ipp = &ip; printf("%p\n", ipp); printf("%p\n", *ipp);
    printf("%d\n", **ipp);
/* Function returning a pointer to void: */
void *fn();
/* Pointer to function returning void: */
void (*fn)(void);
/* Pointer to function taking and returning int: */
int (*fn)(int);
```

C: Function pointers

```c
#include <stdio.h>

int inc(int i)
{
    return ++i;
}

int dec(int i)
{
    return --i;
}

int main(int argc, char *argv[])
{
    int (*f)(int) = inc;
    printf("%d\n", f(2));
}
```

C: Pointer arithmetic

```c
#include <stdio.h>

int main(void)
{
        int i[] = { 0, 1 };
        int *ip = &i[0];

        printf("%d\n", ip[0]);
        printf("%d\n", ip[1]);

        printf("%d\n", *ip);
        printf("%d\n", *(ip + 1));

        printf("%ld\n", (long) ip);
        printf("%ld\n", (long) (ip + 1));
}
```

C: Socket

```c
#include <sys/socket.h>
#include <netdb.h>
#include <unistd.h>

int main(void)
{
  int fd; ssize_t size; char buf[1024];
  struct addrinfo hints = { 0 };
  struct addrinfo *info = NULL, *p;
  getaddrinfo("towel.blinkenlights.nl", "23", &hints, &info);
  for (p = info; p; p = p->ai_next) {
    fd=socket(p->ai_family,p->ai_socktype,0);
    if (fd != -1) {
      connect(fd, p->ai_addr, p->ai_addrlen);
      break;
    }
  } ...
```

C: Socket (continued)

```
1  ...

3    for (;;) {
4      size = recv(fd, buf, 1024, 0);
5      if (0 == size) {
6        break;
7      } else if (size > 0) {
8        write(STDOUT_FILENO, buf, size);
9      } else {
10       break;
11     }
12   }
13 }
```

C: Gets

```c
#include <stdio.h>

int main(void)
{
    char buf[16];

    gets(buf);

    printf("%s\n", buf);
}
```

C: ?

```c
#include <stdio.h>

void foo(void)
{
    int b = 42;
}

void bar(void)
{
    int a;
    printf("%d\n", a);
}

int main(void)
{
    foo();
    bar();
}
```