

C勉強会 第三回

慶應義塾大学大学院
システムデザイン・マネジメント研究科
立山 義祐

ポインタ(1/2)

```
// 3-1.c
#include <stdio.h>

int
main(int ac, char **av)
{
    int a = 10;
    int *p = &a;

    printf("%d\n", *p);
    printf("%d\n", a);

    return 0;
};
```

ポインタ(2/2)

```
// 3-2.c
#include <stdio.h>

int
main(int ac, char **av)
{
    int a = 10;
    int *p = &a;

    printf("%d\n", a);

    *p = 20;

    printf("%d\n", a);

    return 0;
};
```

ポインタと配列

```
// 3-3.c
#include <stdio.h>

int
main(int ac, char **av)
{
    int a[11] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0};
    int i;
    int *p;

    for (i = 0; i < 11; ++i) {
        printf("%d %d\n", i, a[i]);
    }

    for (p = a; 0 != *p; ++p) {
        printf("%d\n", *p);
    }

    return 0;
};
```

ポインタとCの文字列の正体

```
// 3-4.c
#include <stdio.h>
```

```
int
main(int ac, char **av)
{
```

```
    int i;
    char a[5] = "test";
    char *p;

    for (i = 0; i < 5; i++) {
        printf("%d %c %d\n", i, a[i], a[i]);
    }

    for (p = a; 0 != *p; ++p) {
        printf("%c %d\n", *p, *p);
    }
}
```

```
};
```

- Cの文字列は「文字」の「配列」
- 文字列終端を示す文字の文字コードは0

ポインタとCの文字列の正体

```
// 3-5.c
#include <stdio.h>
```

```
void
print_bc(char b[], char c[])
{
    printf("b = %s\n", b);
    printf("c = %s\n", c);
}
```

```
int
main(int ac, char **av)
{
```

```
    int i;
    char a[5] = "test";
    char b[5] = "";
    char c[5] = "";
    char *tgt, *src;
```

```
    print_bc(b, c);
```

```
    for (i = 0; i < 5; ++i) {
        b[i] = a[i];
    }
```

```
    print_bc(b, c);
```

```
    src = a;
    tgt = c;
```

```
    for (; 0 != *src; ++src, ++tgt) {
        *tgt = *src;
    }
    *tgt = 0;
```

```
    print_bc(b, c);
```

```
    return 0;
```

```
};
```

文字列操作のための関数 strcpy(), sprintf()

```
// 3-6.c
#include <stdio.h>
#include <string.h>

int
main(int ac, char **av)
{
    char str[] = "test";
    char a[10] = "";
    char b[100] = "";

    strcpy(a, str);
    sprintf(b, "a is \"%s\"", a);

    printf("%s\n", a);
    printf("%s\n", b);

    return 0;
};
```

文字列入力のための関数 gets()

```
// 3-7.c
#include <stdio.h>

int
main(int ac, char **av)
{
    char buf[100];

    gets(buf);

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

    return 0;
};
```

struct の使い方

```
// 3-8.c
#include <stdio.h>
```

```
struct point {
    char name[30];
    float x, y;
};
```

```
int
main(int ac, char **av)
{
    struct point p1 = {"p1", 1.0f, 2.0f}, p2 = {"p2", 3.0f, 4.0f}, p3;

    sprintf(p3.name, "%s + %s", p1.name, p2.name);
    p3.x = p1.x + p2.x;
    p3.y = p1.y + p2.y;

    printf("%s = (%f, %f)\n", p3.name, p3.x, p3.y);

    return 0;
};
```

関数呼び出しの引数の渡され方

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

```
struct point {
    char name[30];
    float x, y;
};

void
pointer_add(struct point *tgt,
            struct point *s1, struct point *s2)
{
    sprintf(tgt->name, "%s + %s", s1->name, s2->name);
    tgt->x = s1->x + s2->x;
    tgt->y = s1->y + s2->y;
}
```

```
int
main(int ac, char **av)
{
    struct point p1 = {"p1", 1.0f, 2.0f},
                p2 = {"p2", 3.0f, 4.0f}, p3;

    pointer_add(&p3, &p1, &p2);

    printf("%s = (%f, %f)\n", p3.name,
          p3.x, p3.y);

    return 0;
};
```

File I/O (Input / Output)

// 3-10.c

#include <stdio.h>

int

main(int ac, char **av)

{

FILE *fp;

char *fn = "out.txt";

float val;

printf("Input a value: ");

scanf("%f", &val);

fp = fopen(fn, "w");

if (NULL == fp) {

perror(fn);

return 1;

}

fprintf(fp, "%f\n", val);

fclose(fp);

return 0;

}

File I/O (Input / Output)

// 3-11.c

#include <stdio.h>

int

main(int ac, char **av)

{

FILE *fp;

char *fn = "out.txt";

float val;

fp = fopen(fn, "r");

if (NULL == fp) {

perror(fn);

return 1;

}

fscanf(fp, "%f", &val);

fclose(fp);

printf("val: %f\n", val);

return 0;

}

残り

- 文字コードと man ascii
- 値呼び出し、struct 値渡し(結果を貰う方法)
- man strcpy
- #include <string.h>
- fgets(buf, size, stdin);
- man fgets
- sizeof
- 変数のスコープ

残り

- 2d arrays (int?)
- gdb コマンド
- sizeof
- malloc() 関数
- dangling else, switch