

May 11, 04 16:19

eg47-fptr.c

Page 1/1

```

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

int println(char *s)
{
    return printf("%s\n", s);
}

int main()
{
    /*
     * p is a pointer to a function that returns an int
     * *p is a function that returns an int
     * (*p)() is an int
     */

    int (*p)(char *);

    p = println;
    (*p)("hello world");
}

```

May 12, 04 11:05

eg48-pgm.c

Page

```

#include <stdio.h>

typedef struct file_format
{
    char magic_number[2];
    unsigned char ** (*read_func)(FILE *, int *, int *);
    void (*write_func)(FILE *, int, int, unsigned char **);
} file_format;

typedef unsigned char pixel;

unsigned char **read_ascii_pgm(FILE *, int *, int *);
void write_ascii_pgm(FILE *, int, int, pixel**);

unsigned char **read_text_pgm(FILE *, int *, int *);
void write_text_pgm(FILE *, int, int, pixel**);

#define PGM_ASCII 0
#define PGM_BINARY 1

int main()
{
    file_format fmt_info[] = {
        {"P2", read_ascii_pgm, write_ascii_pgm},
        {"P5", read_text_pgm, write_text_pgm},
    };

    FILE *f = fopen("a.pgm", "rb");
    int w, h;
    char c[2];
    file_format *fmt;

    fread(c, 2, 1, f);
    if (c[1] == '2') {
        fmt = &fmt_info[PGM_ASCII];
    } else if (c[1] == '5') {
        fmt = &fmt_info[PGM_BINARY];
    }

    unsigned char **data = (*fmt->read_func)(f, &w, &h);
    // process image here
    (*fmt->write_func)(f, w, h, data);
}

pixel **read_ascii_pgm(FILE *f, int *w, int *h) {}
void write_ascii_pgm(FILE *f, int w, int h, pixel **data) {}

pixel **read_text_pgm(FILE *f, int *w, int *h) {}
void write_text_pgm(FILE *f, int w, int h, pixel **data) {}

```

May 12, 04 11:06

eg49-typedef.c

Page 1/1

```

#include <stdio.h>

typedef unsigned char pixel;

typedef pixel **(*reader)(FILE *, int*, int*);
typedef void (*writer)(FILE *, int, int, pixel **);

typedef struct file_format
{
    char magic_number[2];
    reader read_func;
    writer write_func;
} file_format;

pixel **read_binary_pgm(FILE *, int *, int *);
void write_binary_pgm(FILE *, int, int, pixel **);
pixel **read_ascii_pgm(FILE *, int *, int *);
void write_ascii_pgm(FILE *, int, int, pixel **);

#define PGM_ASCII 0
#define PGM_BINARY 1

int main()
{
    file_format fmt_info[] = {
        {"P2", read_ascii_pgm, write_ascii_pgm},
        {"P5", read_binary_pgm, write_binary_pgm},
    };

    FILE *f = fopen("a.pgm", "rb");
    int w, h;
    char c[2];
    file_format *fmt;

    fread(c, 2, 1, f);
    if (c[1] == '2') {
        fmt = &fmt_info[PGM_ASCII];
    } else if (c[1] == '5') {
        fmt = &fmt_info[PGM_BINARY];
    }

    pixel **data = (*fmt->read_func)(f, &w, &h);
    // process image here
    (*fmt->write_func)(f, w, h, data);
}

pixel **read_ascii_pgm(FILE *f, int *w, int *h) {}
void write_ascii_pgm(FILE *f, int w, int h, pixel **data) {}
pixel **read_binary_pgm(FILE *f, int *w, int *h) {}
void write_binary_pgm(FILE *f, int w, int h, pixel **data) {}

```

Wednesday May 12, 2004

May 12, 04 11:07

eg50-qsort.c

Page

```

#include <stdlib.h>

typedef struct room {
    char building[5];
    int level;
    int number;
} room;

int roomcmp(const void *r1, const void *r2)
{
    room *room1 = (room *)r1;
    room *room2 = (room *)r2;

    int order = strcmp(room1->building, room2->building);
    if (order != 0)
        return order;
    else {
        if (room1->level < room2->level)
            return -1;
        else if (room1->level > room2->level)
            return 1;
        else {
            if (room1->number < room2->number)
                return -1;
            else if (room1->number > room2->number)
                return 1;
            else
                return 0;
        }
    }
}

int main()
{
    room r[] = {
        {"S16", 3, 12}, {"S15", 1, 2},
        {"SOC1", 4, 20}, {"S16", 3, 10},
        {"S15", 2, 9}, {"SOC1", 8, 17},
    };
    int i;
    qsort(r, sizeof(r)/sizeof(room), sizeof(room), roomcmp);

    for (i = 0; i < sizeof(r)/sizeof(room); i++) {
        printf("%s #%02d-%#02d\n",
            r[i].building, r[i].level, r[i].number);
    }
    return 0;
}

```

eg49-typedef.c, eg50-qsort.c

May 11, 04 17:55

eg51-timer.c

Page 1/1

```
#include <time.h>

typedef void (*timer_callback)(void *);

void do_every(int period, timer_callback what_to_do, void *data)
{
    while (1) {
        what_to_do(data);
        sleep(period);
    }
}

void print_time(void *data)
{
    static int i = 0;
    printf("%3d\n", i++);
}

int main()
{
    do_every(3, print_time, NULL);
}
```

May 12, 04 10:24

eg52-vararg.c

Page

```
#include <stdarg.h>

int sum(int i, ...)
{
    int sum = i;
    int value = -1;
    va_list vp;
    va_start(vp, i);
    while (value != 0)
    {
        value = va_arg(vp, int);
        sum += value;
    }
    va_end(vp);
    return sum;
}

int main()
{
    printf("%d\n", sum(1, 2, 3, 4, 5, 0));
    printf("%d\n", sum(10, 11, 12));
}
```