

```
// Programming 1. Sample at 2017-04-13-13-54.09
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

const size_t BUFFERSZ = 1000;
int g_location_items = 0;

struct LocationIterator
{
    long    filepos;
    char    location[20];
    struct LocationIterator *next;
};

struct LocationIterator *locations = NULL;
struct LocationIterator **nextLocationItem = &locations;

// shifts <location> to the beginning of the buf
void shift_buffer(char *buf, size_t size, size_t tailsz, FILE *file)
{
    memmove(buf, buf+size, (BUFFERSZ-1)-size);
    int rsz = fread(buf+((BUFFERSZ-1)-size), 1, size, file);
    buf[rsz]=0;
}

void extractLocation(const char *src, char *dest)
{
    memset(dest, 0, 20);
    const char * endTag = strstr(src, "<");
    if(NULL == endTag || (endTag - src)>19)
        strncpy(dest, src, 19);
    else
        strncpy(dest, src, endTag-src);
}

// add new LocationItem
void addLocation(const char *tag, FILE *file)
{
    const char *locationValue = tag+strlen("<location>");

    *nextLocationItem =
        (struct LocationIterator*)malloc(sizeof(struct LocationIterator));

    (*nextLocationItem)->filepos = ftell(file)-(BUFFERSZ-1);

    extractLocation(locationValue, (*nextLocationItem)->location);

    printf("%d: Location: [%s] \n", g_location_items++,
        (*nextLocationItem)->location);
}
```

```
*nextLocationItem=(*nextLocationItem)->next;

}

// Search the next <location> tag
void find_all_locations(FILE *file, char *buf)
{
    char *startTag = NULL;
    char *locationPtr = NULL;
    do
    {
        size_t rd = fread(buf, 1, BUFFERSZ-1, file);
        if(rd <= 1)
            break;

        startTag=strchr(buf, '<');

        // read the next buffer, there's no <
        if( NULL == startTag)
            continue;

        shift_buffer(buf, startTag-buf, startTag-buf, file);

        locationPtr = strstr(buf, "<location>");

        if(buf == locationPtr)
            addLocation(buf, file);

    } while(!feof(file));
}

void initLocations(const char *filename)
{
    char buf[BUFFERSZ];

    FILE *file = fopen(filename, "r");
    printf("file = %p\n", file);

    if(NULL == file)
        return;

    memset(buf, 0, BUFFERSZ);
    find_all_locations(file, buf);
    fclose(file);
}

int main(/*int c, char **v */)
{
```

```
initLocations("./database");  
return 0;  
}
```

From:

<https://se.moevm.pro/> - **МОЭВМ Вики** [se.moevm.pro]

Permanent link:

<https://se.moevm.pro/doku.php/start:prog2:lectures:code4>

Last update:

