プログラミングII

第7回: 動的メモリ

2019年1月9日(水)

筑波大学 情報メディア創成学類
三河 正彦

文字列の操作

- 教科書10.4 文字列の長さを実行時に決める.

メモリの動的割り当て

```
/* static.c */
#include <stdio.h>

#define BUFSIZE 200

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

  printf("Input name => ");
  fgets(buf, BUFSIZE, stdin);

  printf("Name is %s", buf);
}
```

```
/* dynamic.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define BUFSIZE 200

int main(void)
{
  char buf[BUFSIZE];
  char *name;

  printf("Input name => ");
  fgets(buf, BUFSIZE, stdin);
  name = (char *)malloc(sizeof(char)
                  * (strlen(buf)+1));
  strcpy(name, buf);

  printf("Name is %s", name);
  free(name);
}
```

static3.c

- 5人分の名前を受け取れるようにするため，bufの宣言を2次元配列の宣言に変更する.
- 次に，キーボードから入力された名前(文字列)を受け取る部分と，表示する部分を繰り返しを用いて，5人分の処理ができるようにする.
- また，このプログラムが使用するメモリ量を表示する部分を作成し，実行結果を確認せよ.
- ソースファイル名はstatic3.cとする.

static3.c

```
 1 /* static3.c */
 2 #include <stdio.h>
 3
 4 #define BUFSIZE 200
 5 #define NUM     5
 6
 7 int main(void)
 8 {
 9   char buf[NUM][BUFSIZE];
10   int count;
11
12   for (count=0; count<NUM; count++) {
13     printf("Input %dth name => ", count+1);
14     fgets(buf[count], BUFSIZE, stdin);
15   }
16
17   for (count=0; count<NUM; count++) {
18     printf("%dth name: %s", count+1, buf[count]);
19   }
20
21   printf("sizeof(buf) = %d\n", sizeof(buf));
22 }
```

実行例

```
# ./a.out
Input 1th name => mikawa1
Input 2th name => mikawa2
Input 3th name => mikawa3
Input 4th name => mikawa4
Input 5th name => mikawa5
1th name: mikawa1
2th name: mikawa2
3th name: mikawa3
4th name: mikawa4
5th name: mikawa5
sizeof(buf) = 1000
#
```

static4.c

- static3.cを改造し，５人分の名前を格納する変数として，２次元配列ではなく，構造体を使用して実現する.
- 構造体の形態は，各自考えること.
- ソースファイル名はstatic4.cとする.

static4.c

```
 1 /* static4.c */
 2 #include <stdio.h>
 3
 4 #define BUFSIZE 200
 5 #define NUM     5
 6
 7 struct user {
 8     char name[BUFSIZE];
 9 };
10
11 int main(void)
12 {
13   int i;
14   struct user person[NUM];
15
16   for (i=0; i<NUM; i++) {
17     printf("No. %d\n", i);
18     printf("Input name => ");
19     fgets(person[i].name, BUFSIZE, stdin);
20   }
21
22   for (i=0; i<NUM; i++) {
23     printf("Name is %s", person[i].name);
24   }
25
26     printf("sizeof(person) = %d\n", sizeof(person));
27 }
```

実行例

```
# ./a.out
No. 0
Input name => mikawa1
No. 1
Input name => mikawa2
No. 2
Input name => mikawa3
No. 3
Input name => mikawa4
No. 4
Input name => mikawa5
Name is mikawa1
Name is mikawa2
Name is mikawa3
Name is mikawa4
Name is mikawa5
sizeof(person) = 1000
#
```

static4.c (その2)

```
 1 /* static4.c */
 2 #include <stdio.h>
 3
 4 #define BUFSIZE 200
 5 #define NUM     5
 6
 7 typedef struct FiveNames {
 8     char name1[BUFSIZE];
 9     char name2[BUFSIZE];
10     char name3[BUFSIZE];
11     char name4[BUFSIZE];
12     char name5[BUFSIZE];
13 } FiveNames;
14
15 int main(void)
16 {
17     FiveNames names;
18
19     printf("Input 1st name => ");
20     fgets(names.name1, BUFSIZE, stdin);
21     printf("Input 2nd name => ");
22     fgets(names.name2, BUFSIZE, stdin);
23     printf("Input 3rd name => ");
24     fgets(names.name3, BUFSIZE, stdin);
25     printf("Input 4th name => ");
26     fgets(names.name4, BUFSIZE, stdin);
27     printf("Input 5th name => ");
28     fgets(names.name5, BUFSIZE, stdin);
29
30     printf("1st Name is %s", names.name1);
31     printf("2nd Name is %s", names.name1);
32     printf("3rd Name is %s", names.name1);
33     printf("4th Name is %s", names.name1);
34     printf("5th Name is %s", names.name1);
35
36     printf("sizeof(names) = %d\n",
37            sizeof(names));
38 }
```

static4.c (その2)

```
 1 /* static4.c */
 2 #include <stdio.h>
 3
 4 #define BUFSIZE 200
 5 #define NUM     5
 6
 7 typedef struct FiveNames {
 8     char name1[BUFSIZE];
 9     char name2[BUFSIZE];
10     char name3[BUFSIZE];
11     char name4[BUFSIZE];
12     char name5[BUFSIZE];
13 } FiveNames;
14
15 int main(void)
16 {
17     FiveNames names;
18
```

```
19     printf("I
20     fgets(nam
21     printf("I
22     fgets(nam
23     printf("I
24     fgets(nam
25     printf("I
26     fgets(nam
27     printf("I
28     fgets(nam
29
30     printf("1
31     printf("2
32     printf("3
33     printf("4th Name is %s", names.name1);
34     printf("5th Name is %s", names.name1);
35
36     printf("sizeof(names) = %d\n",
37            sizeof(names));
38 }
```

実行例

```
# ./a.out
Input 1st name => mikawa1
Input 2nd name => mikawa2
Input 3rd name => mikawa3
Input 4th name => mikawa4
Input 5th name => mikawa5
1st Name is mikawa1
2nd Name is mikawa1
3rd Name is mikawa1
4th Name is mikawa1
5th Name is mikawa1
sizeof(names) = 1000
#
```

動的なメモリ割当: dynamic2.c

- dynamic.c(p.3)では，メモリ確保の際にエラーチェックをしていない．これをエラーチェックをするようにせよ．
- malloc()の返り値がNULL だったら，エラーメッセージとしてmemory allocation error!!と表示して，プログラムを終了するように変更せよ．
- ソースファイル名をdynamic2.cとする．
- プログラムの終了は関数exit()を使う．

dynamic2.c

```
 1 /* dynamic2.c */
 2 #include <stdio.h>
 3 #include <stdlib.h>
 4 #include <string.h>
 5
 6 #define BUFSIZE 200
 7
 8 int main(void)
 9 {
10     char buf[BUFSIZE];
11     char *name;
12
13     printf("Input your name => ");
14     fgets(buf, BUFSIZE, stdin);
15
16     name = (char *)malloc(sizeof(char)
17             * (strlen(buf)+1));
18     if (name == NULL) {
19         printf("memory allocation error!!");
20         exit(1);
21     }
22
23     strcpy(name, buf);
24
25     printf("Your name : %s", name);
26
27     free(name);
28 }
```

メモリ領域

- スタック領域: ローカル変数や関数の引数等が利用する領域
  - 容量は小さい(数MB程度) -> 大きな配列等の確保が困難
- ヒープ領域: malloc()等を用いて動的に確保する領域
  - 容量が大きい -> 大きな配列等を確保するのに適している
- 静的領域: グローバル変数やstatic変数が利用する領域
  - プログラム実行時に確保される固定のサイズ
- テキスト領域: 機械語に翻訳されたプログラムが格納される領域

```
micPRO[133] ulimit -a
 中略...
open files                          (-n) 256
pipe size               (512 bytes, -p) 1
stack size                 (kbytes, -s) 8192
cpu time                  (seconds, -t) unlimited
max user processes                  (-u) 709
virtual memory             (kbytes, -v) unlimited
```

メモリ領域

```
#include <stdio.h>

#define MAX 100

typedef struct matrix {
    int rows;
    int columns;
    double data[MAX][MAX];
} matrix;
```

1000だと実行時にエラー

```
int main(void)
{
    int i, j, k;
    struct matrix X[100];

    for (i=0; i<100; i++) {
      for (j=0; j<MAX; j++) {
        for (k=0; k<MAX; k++) {
          printf("X[%d].data[%d][%d] = %lf\n",
                 i, j, k, X[i].data[j][k]);
        }
      }
    }
}
```

おわり

筑波大学情報メディア創成学類 三河 正彦