



2007年10月31日

## 第11回 C言語によるプログラミング (8)

# 情報処理論 (応用)

松山大学 経営学部

檀 裕也

<http://www.cc.matsuyama-u.ac.jp/~dan/education/application/>



# 前回の実習課題

- 入力した文字列のうち、アルファベット小文字を大文字に変換して表示するプログラム `strconvert.c` を作成せよ。
  - 宛先: dan@cc.matsuyama-u.ac.jp
  - 件名: プログラミング課題#09

```
C:¥>strconvert.exe  
文字列:Hello_world!  
HELLO_WORLD!  
  
C:¥>
```



# 解答例

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

```
strconvert.c
```

```
int main()
```

```
{
```

```
    int i;
```

```
    char str[20];
```

```
    printf( "文字列:" );
```

```
    scanf( "%s", str );
```

```
    for( i = 0; i < 20 && str[i] != '¥0'; i++ ){  
        if( 0x61 <= str[i] && str[i] <= 0x7A ){  
            printf( "%c", str[i] - 0x20 );  
        }  
        else{  
            printf( "%c", str[i] );  
        }  
    }
```

```
    return 0;
```

```
}
```

# マクロ

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

```
#define LENGTH 40
```

```
int main()
```

```
{
```

```
    int i;
```

```
    char str[LENGTH];
```

```
    printf( "文字列：" );
```

```
    scanf( "%s", str );
```

```
    for( i = 0; i < LENGTH && str[i] != '\0'; i++ ){
```

```
        if( 0x61 <= str[i] && str[i] <= 0x7A ){
```

```
            printf( "%c", str[i] - 0x20 );
```

```
        }
```

```
        else{
```

```
            printf( "%c", str[i] );
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

ソースコード中の  
LENGTH をすべて 40 に  
置き換える



# 今回の授業内容

---

- C言語によるプログラミング (8)
  - 配列
  - ポインタ



# 例題 1

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

```
init_int.c
```

```
int main()
```

```
{
```

```
    int i;
```

```
    int num[] = { 2007, 10, 31, 16, 15 };
```

```
    for( i = 0; i < 5; i++ ){  
        printf( "%d¥n", num[i] );
```

```
    }
```

```
    return 0;
```

```
}
```



# 配列の初期化

- 配列の宣言と同時に、要素に値を代入する

```
int num[] = { 2007, 10, 31, 16, 15 };
```

```
int num[5];  
num[0] = 2007;  
num[1] = 10;  
num[2] = 31;  
num[3] = 16;  
num[4] = 15;
```

```
C:¥>init_int.exe
```

```
2007
```

```
10
```

```
31
```

```
16
```

```
15
```

```
C:¥>
```



## 例題 2

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

```
init_str.c
```

```
int main()  
{  
    int i;  
    char str[] = "Hello, world¥n";  
  
    for( i = 0; str[i] != '¥0'; i++ ){  
        printf( "%c %08x¥n", str[i], &str[i] );  
    }  
  
    return 0;  
}
```





# 配列の初期化

- 文字列は、より簡単に初期化できる

```
char str[] = "Hello, world¥n";
```

```
char str[??];  
str[0] = 'H';  
str[1] = 'e';  
str[2] = 'l';  
str[3] = 'l';  
str[4] = 'o';
```

```
C:¥>init_str.exe  
H 0012ff58  
e 0012ff59  
l 0012ff5a  
:  
d 0012ff63  
  
C:¥>
```



# 解説

- アドレス演算子

- 変数や配列の要素に & をつけると、そのメモリ上のアドレスを示す

H
e
l
l
o

アドレス	データ
00120000	'H'
00120001	'e'
00120002	'l'
00120003	'l'
00120004	'o'



# ポインタ

- 変数などのアドレスを格納するための変数
  - そのアドレスに格納されている値を参照できる

```
char str[] = "ABC"; // メモリ上に "ABC" を配置
char *p; // ポインタ変数 p を宣言

p = str; // p に str のアドレスを代入
p++; // 次のアドレスに移動する

printf( "%c", *p ); // アドレス p の文字を表示
```



# 例題 3

```
#include <stdio.h>

int main()
{
    int i;
    char str[] = "Hello, world!";
    char *p;

    p = str;

    while( *p != '\0' ){
        p++;
    }

    p--;

    while( str <= p ){
        printf( "%c", *p );
        p--;
    }
}
```

```
        return 0;
    }
```

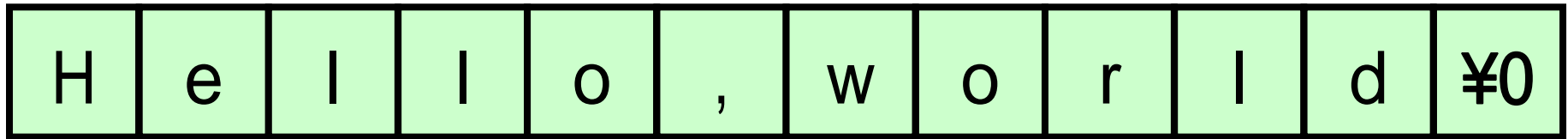
**str\_reverse.c**



# 実行結果

- 与えられた文字列を逆順に表示する

配列 **str**



↑  
ポインタ **p**

```
C:¥>str_reverse.exe  
!dlrow ,olleH  
C:¥>
```



# 例題 4

main.c

```
#include <stdio.h>

int main( int argc, char *argv[] )
{
    int i;

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

    return 0;
}
```



# 実行結果

- プログラム起動時に与えた引数を表示する
  - int argc                    引数の個数
  - char \*argv[]                引数の内容 (文字列)

```
C:¥>main.exe Hello, world!  
main.exe  
Hello,  
world!  
  
C:¥>
```



# 実習課題

- 次の問題に解答せよ。
  - 例題2における配列 `str` の要素数を答えよ。また、その内容を簡単に説明せよ。
  - ポインタとは何か説明せよ。
- 宛先: `dan@cc.matsuyama-u.ac.jp`
- 件名: プログラミング課題#10





# 次回の予定

---

- 日時： 11月 2日 (金)  
4 時限
- OpenGL によるコンピュータグラフィックス  
– 基礎知識