ソフトウェア概論 A/B (2019/06/14)

Ver. 1.0

栗野 俊一
kurino@math.cst.nihon-u.ac.jp
http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino/2019/soft/soft.html

2019 年 6 月 14 日

概 要

ソフトウェア概論 A/B[1] の 2019 年 6 月 14 日 の資料 [2]

目 次

[1] http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino/2019/soft/soft.html
[2] http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino/2019/soft/20190614/20190614.html

1 講義資料

1.1 当日の OHP 資料

- 当日の OHP 資料です。
  - 表示用 (HTML)[3]
  - 印刷用 (PDF)[4]

1.2 講義で利用するサンプルプログラム

Download : sample-001.c[5]

リスト 1: sample-001.c

```
/*
 * 2019/06/14 sample-001.c
 */

/*
 * "Hello, World" again
 */

#include <stdio.h>

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

  printf ( "Hello, World\n" );        /* "Hello, World" という文字列と改行 ( "\n" ) を表示
する */

  return 0;
}
```

```
$ ./sample-001.exe
Hello, World
$
```

図 1: sample-001.c の実行結果

Download : sample-002.c[6]

リスト 2: sample-002.c

[3] ohp/html/index.html
[4] ohp/ohp.pdf
[5] program/sample-001.c
[6] program/sample-002.c

```
/*
 * 2019/06/14 sample-002.c
 */

/*
 * プログラム・ノックアウト (1)
 */

#include <stdio.h>

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

        /* まず、printf をノックアウト */

  return 0;
}
```

```
$ ./sample-002.exe
$
```

図 2: sample-002.c の実行結果

Download : sample-003.c[7]

リスト 3: sample-003.c

```
/*
 * 2019/06/14 sample-003.c
 */

/*
 * プログラム・ノックアウト (2)
 */

/*
 * main 関数もなくしたら ?
 */

/* ここには何もない */
```

Download : sample-004.c[8]

リスト 4: sample-004.c

[7] program/sample-003.c
[8] program/sample-004.c

```
$ cc -c sample-003.c
$ cc -o sample-003.exe sample-003.o
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 0 が無効なシンボル索引 11 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 1 が無効なシンボル索引 12 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 2 が無効なシンボル索引 2 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 3 が無効なシンボル索引 2 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 4 が無効なシンボル索引 11 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 5 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 6 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 7 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 8 が無効なシンボル索引 12 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 9 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 10 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 11 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 12 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 13 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 14 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 15 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 16 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 17 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 18 が無効なシンボル索引 13 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): 再 配
置 19 が無効なシンボル索引 21 を持っています
/usr/bin/ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_line): 再 配
置 0 が無効なシンボル索引 2 を持っています
/usr/lib/gcc/x86_64-linux-gnu/4.8/../../../x86_64-linux-gnu/crt1.o: 関
数 `_start' 内:
(.text+0x20): `main' に対する定義されていない参照です
collect2: error: ld returned 1 exit status
$
```

```
/*
 * 2019/06/14 sample-004.c
 */

/*
 * プログラム・ノックアウト (3)
 */

/*
 *        #include <stdio.h> // #include を無くした
 */

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

  printf ( "Hello, World\n" );        /* この行で「警告」が出る */

  return 0;
}
```

```
$ cc -c sample-004.c
sample-004.c: In function ‘ main ’:
sample-004.c:15:3: warning: incompatible implicit declaration of built-in function ‘ printf ’
[enabled by default]
   printf ( "Hello, World\n" ); /* この行で「警告」が出る */
   ^
$
```

図 4: sample-004.c の実行結果

Download : sample-005.c[9]

リスト 5: sample-005.c

```
/*
 * 2019/06/14 sample-005.c
 */

/*
 * プログラム・ノックアウト (4)
 */

/*
 *        #include <stdio.h> // #include を無くした
 */

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

        /* printf も使わないと、「警告」はなくなった */

  return 0;
}
```

[9] program/sample-005.c

```
$ ./sample-005.exe
$
```

図 5: sample-005.c の実行結果

Download : sample-006.c[10]

リスト 6: sample-006.c

```
/*
 * 2019/06/14 sample-006.c
 */

/*
 * プログラム・ノックアウト (5)
 */

void main ( void ) {        /* int を void に変更, 引数宣言もなくして良い */

         /* return 0; を削って、空っぽにする */
}

/*
        最も短い C 言語プログラムの実現
*/
```

```
$ ./sample-006.exe
$
```

図 6: sample-006.c の実行結果

Download : sample-007.c[11]

リスト 7: sample-007.c

```
/*
 * 2019/06/14 sample-007.c
 */

/*
 * プログラム・ノックアウト (6)
 */

/*
```

[10] program/sample-006.c
[11] program/sample-007.c

```
 *        #include の代りに printf の extern 宣言
 */

extern int printf ( const char *message, ... );

        /* 「extern void printf ( char *message );」 ではない ?? */

/*
 * main
 */

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

  printf ( "Hello, World\n" );        /* 「警告」が消えた */

  return 0;
}
```

```
$ ./sample-007.exe
Hello, World
$
```

図 7: sample-007.c の実行結果

Download : sample-008.c[12]

リスト 8: sample-008.c

```
/*
 * 2019/06/14 sample-008.c
 */

#include <stdio.h>

/*
 * 文字列を三度表示する : 関数定義の例
 */

void threeTimesPrint ( char *message )        {

         printf ( message );                /* 一度目 */
         printf ( message );                /* ニ度目 */
         printf ( message );                /* 三度目 */

}

/*
 * main
 */

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

[12] program/sample-008.c

```
        threeTimesPrint ( "Hello\n" );                  /* 関数呼出しの例 */

        threeTimesPrint ( "こんにちは\n" );

        return 0;
}
```

```
$ ./sample-008.exe
Hello
Hello
Hello
こんにちは
こんにちは
こんにちは
$
```

図 8: sample-008.c の実行結果

Download : sample-009.c[13]

リスト 9: sample-009.c

```
/*
 * 2019/06/14 sample-009.c
 */

#include <stdio.h>

/*
 * echo_char ( char ch )
 *        char ch        --        引数で指定された文字
 *                引数で指定された文字を『』付で出力
 */

void echo_char ( char ch )        {

         printf ( "『" );
         putchar ( ch );                /* 引数で指定された文字を出力する */
         printf ( "』" );
}

/*
 * main
 */

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

        printf ( "何か文字を入れてから [Enter] キーを押す : " );
```

[13] program/sample-009.c

```
        echo_char ( getchar() );        /* getchar() で文字を読込み、それを出力 */
        putchar ( '\n' );

        return 0;
}
```

```
A
```

図 9: 入力例

```
$ ./sample-009.exe < sample-009.in
何か文字を入れてから [Enter] キーを押す : A
『A』
$
```

図 10: sample-009.c の実行結果

Download : sample-010-01.c[14]

リスト 10: sample-010-01.c

```
/*
 * 2019/06/14 sample-010-01.c
 */

#include <stdio.h>                /* putchar が必要なので.. */

/*
 *        nprinttail
 *        n 回数 message を出力するが、行末を変更する。
 */

void nprinttail ( char *tail, char *message ) {

        if ( *tail == '\0' ) {
        } else {
                printf ( message );
                if ( *tail == '\n' )        {        /* 改行の時のみ */
                        putchar ( '\n' );                /* 改行する */
                } else {                                        /* それ以外は */
                          /* 何もしない */
                }

                nprinttail ( tail + 1, message );
        }
}
```

[14] program/sample-010-01.c

Download : sample-010.c[15]

リスト 11: sample-010.c

```
/*
 * 2019/06/14 sample-010.c
 */

/*
 * extern 宣言
 */

extern void println ( char *message );
extern void nprinttail ( char *tail, char *message );

/*
 * n 回数の繰返し
 */

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

        println ( "毎回改行「Hello」: " );
        nprinttail ( "\n\n\n", "Hello" );

        println ( "3 回目, 7 回目だけ改行「Hello」: " );
        nprinttail ( "12\n456\n", "Hello" );

        return 0;
}
```

```
$ ./sample-010.exe
$
```

図 11: sample-010.c の実行結果

Download : sample-011.c[16]

リスト 12: sample-011.c

```
/*
 * 2019/06/14 sample-011.c
 */

#include <stdio.h>

/*
 * 文字の扱い
 */
```

[15] program/sample-010.c
[16] program/sample-011.c

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

        printf ( "abc\n" );                /* 「abc(改行)」とでる */
        printf ( "abc\n" + 1 );         /* 「bc(改行)」とでる ( 1 を加えると短くなる ) */

        putchar ( 'A' );                /* 「A」とでる */
        putchar ( '\n' );                /* 「(改行)」とでる */

        putchar ( 'A' + 1 );         /* 文字に 1 を加えると ? */
        putchar ( '\n' );                /* 「(改行)」とでる */

        return 0;
}
```

```
$ ./sample-011.exe
abc
bc
A
B
$
```

図 12: sample-011.c の実行結果

Download : sample-012.c[17]

リスト 13: sample-012.c

```
/*
 * 2019/06/14 sample-012.c
 */

/*
 * extern 宣言
 */

extern void printcharln ( char ch );

/*
 * 文字の扱い (2)
 */

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

        printcharln ( 'A' + 1 );              /* 'B' になった */
        printcharln ( 'B' + 1 );              /* 'C' になる */
        printcharln ( 'A' + 1 + 1 );        /* これも 'C' になる */

        printcharln ( 'A' + 0 );              /* これはもちろん 'A' */
        printcharln ( 'A' + 10 );              /* 'A', 'B', .., 'J', 'K' になるはず */
```

[17] program/sample-012.c

```
        printcharln ( 'A' + 25 );               /* 'Z' !! */

        printcharln ( 'Z' + 1 );               /* ??? */

        return 0;
}
```

```
$ ./sample-012.exe
$
```

図 13: sample-012.c の実行結果

Download : sample-013.c[18]

リスト 14: sample-013.c

```
/*
 * 2019/06/14 sample-013.c
 */

/*
 * extern 宣言
 */

extern void nprintcharln ( char *n, char ch );
extern void printcharln ( char ch );
extern void println ( char *string );

/*
 * 文字の扱い (3)
 */

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

        println ( "'A' から 10 個 :" );
        nprintcharln ( "1234567890", 'A' );

        println ( "'U' から 10 個 :" );
        nprintcharln ( "1234567890", 'U' );

        println ( "'k' から 10 個 :" );
        nprintcharln ( "1234567890", 'k' );

        println ( "'0' から 10 個 :" );
        nprintcharln ( "1234567890", '0' );

        println ( "'9' + 1 は.. :" );
        printcharln ( '9' + 1 );               /* 残念ながら "10" ではない */

        return 0;
}
```

[18] program/sample-013.c

```
$ ./sample-013.exe
$
```

図 14: sample-013.c の実行結果

Download : sample-014.c[19]

リスト 15: sample-014.c

```
/*
 * 2019/06/14 sample-014.c
 */

/*
 * extern 宣言
 */

extern void println ( char *string );
extern void printonedigit ( int n );
extern void printonedigitln ( int n );

/*
 * 整数
 */

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

        println ( "整数値 0 の出力" );
        printonedigitln ( 0 );                          /* 「'0'」 でも 「"0"」 でもなく 「0」
*/

        println ( "整数値 9 の出力" );
        printonedigitln ( 9 );                          /* 一桁は OK */

        println ( "整数値 11 の出力" );
        printonedigitln ( 11 );                     /* 上手く行かない */

        println ( "整数値 1 + 1 の出力" );
        printonedigitln ( 1 + 1 );                  /* やっと計算がでてきた */

        println ( "整数値 5 - 2 の出力" );
        printonedigitln ( 5 - 2 );                  /* 引算 */

        println ( "整数値 3 * 2 の出力" );
        printonedigitln ( 3 * 2 );                  /* かけ算 */

        println ( "整数値 8 / 3 の出力" );
        printonedigitln ( 8 / 3 );                   /* 小数点以下は余りは切り捨て */

        println ( "整数値 8 % 3 の出力" );
```

[19] program/sample-014.c

```
        printonedigitln ( 8 % 3 );                  /* 余りは「%」をで計算 */

        println ( "整数値 8 - ( 8 / 3 ) * 3 の出力" );
        printonedigitln ( 8 - ( 8 / 3 ) * 3 );                  /* 余りを求めるもう一つの方
法 */

        return 0;
}
```

```
$ ./sample-014.exe
$
```

図 15: sample-014.c の実行結果

Download : sample-015.c[20]

リスト 16: sample-015.c

```
/*
 * 2019/06/14 sample-015.c
 */

/*
 * extern 宣言
 */

extern void println ( char *string );
extern void printpositiveint ( int n );
extern void printpositiveintln ( int n );

/*
 * 整数
 */

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

        println ( "整数値 0 の出力" );
        printpositiveintln ( 0 );

        println ( "整数値 11 の出力" );
        printpositiveintln ( 11 );

        println ( "整数値 12345 の出力" );
        printpositiveintln ( 12345 );

        println ( "整数値 12 * 34 の出力" );
        printpositiveintln ( 12 * 34 );

        return 0;
}
```

```
$ ./sample-015.exe
$
```

図 16: sample-015.c の実行結果

Download : sample-016.c[21]

リスト 17: sample-016.c

```
/*
 * 2019/06/14 sample-016.c
 */

#include <stdio.h>

/*
 * check_star_char
 */

void check_star_char ( char ch ) {

        if ( ch == '*' ) {                /* ch の中身が '*' かどうかをチェック */
                printf ( "これは「＊」です\n" );
        } else {
                printf ( "これは「＊」ではありません\n" );
        }

}

/*
 * main
 */

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

        check_star_char ( getchar() );        /* getchar() で文字を読込み、それを出力 */

        return 0;
}
```

```
A
```

図 17: 入力例

[20] program/sample-015.c
[21] program/sample-016.c

```
$ ./sample-016.exe < sample-016.in
A
これは「＊」ではありません
$
```

図 18: sample-016.c の実行結果

1.3 講議中に作成したプログラム

- 講議中に作成したプログラム [22]

1.4 本日の課題

課題プログラム内の「/*名前:ここ*/」の部分を書き換え「/*この部分を完成させなさい*/」の部分にプログラムを追加して、プログラムを完成させます。

1.4.1 課題 20190614-01 : キーボードから一文字入力し、その文字によって異る国の挨拶をする

Download : 20190614-01.c[23]

リスト 18: 20190614-01.c

```
/*
 * CDATE FILENAME
 *
 *        キーボードから一文字入力し、その文字によって異る国の挨拶をする
 */

#include <stdio.h>

/*
 * hello ( char contry )
 *        char contry : 国を表す一文字
 *                j : 日本
 *                e : 英語
 *                c : 中国
 *                f : フランス語
 *                g : ドイツ語
 */

void hello ( char cmd ) {

        if ( cmd == 'j' ) {                             /* 'j' の時は、日本語にする */
                printf ( "こんにちは\n" );
        } else if ( cmd == 'e' )        {       /* 'e' の時は、英語にする */

        /*
        **          この部分を完成させなさい
```

[22] p/
[23] ex/01/q/01.c

```
        */

        } else if ( cmd == 'c' )        {       /* 'c' の時は、中国語にする */
                printf ( "ニイハオ\n" );
        } else if ( cmd == 'f' )        {       /* 'f' の時は、フランス語にする */
                printf ( "Bonjour\n" );

        /*
        **          この部分を完成させなさい
        */

                printf ( "Guten tag\n" );
        } else {                                                /* どれでもなければ.. */
                printf ( "???\n" );
        }

}


/*
 *        main
 */

int main( void )
{

        printf ( "国を表す文字を入力してください\n" );
        printf ( "\tj\t 日本\n" );
        printf ( "\te\t 英語\n" );
        printf ( "\tc\t 中国\n" );
        printf ( "\tf\t フランス\n" );
        printf ( "\tg\t ドイツ\n" );

        hello ( getchar() );        /* getchar() で文字を入力し、それに対応する結果を出す */

         return 0;
}
```

```
f
```

図 19: 入力例

1.4.2 課題 20190614-02 : キーボードから一行 (改行まで..) の文字列を読込み、それを逆順に出力する

Download : 20190614-02.c[24]

リスト 19: 20190614-02.c

[24] ex/02/q/02.c

```
$ ./20190614-01-QQQQ.exe
国を表す文字を入力してください
        j       日本
        e       英語
        c       中国
        f       フランス
        g       ドイツ
f
Bonjour
$
```

図 20: 20190614-01.c の実行結果

```
/*
 * CDATE FILENAME
 *
 *        キーボードから一行 (改行まで..) 文字列を読込み、それを逆順に出す
 */

#include <stdio.h>

/*
 * reverse ( char contry )
 *        char cmd : どのメッセージにするかを表す文字
 *
 */

void reverse_line ( char ch ) {

        if ( ch == '\n' )       {                       /* 改行ならば.. */
                /* なにもしなくてよい */
        } else {                                                /* そうでなければ.. */

        /*
        **          この部分を完成させなさい
        */

        }
}


/*
 *        main
 */

int main( void )
{

        reverse_line ( getchar() );         /* 最初の文字を読み込んで .. */
        putchar ( '\n' );                           /* 改行を出力 */

        return 0;
}
```

```
abc123
```

図 21: 入力例

```
$ ./20190614-02-QQQQ.exe
abc123
321cba
$
```

図 22: 20190614-02.c の実行結果