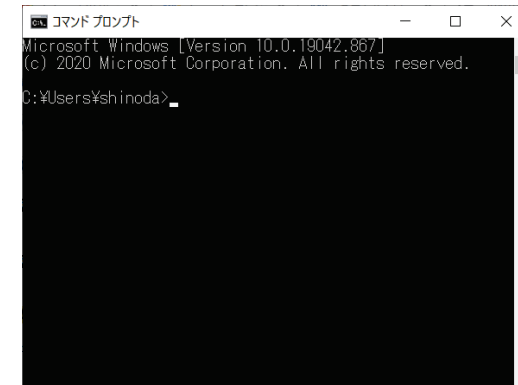


コマンドライン引数

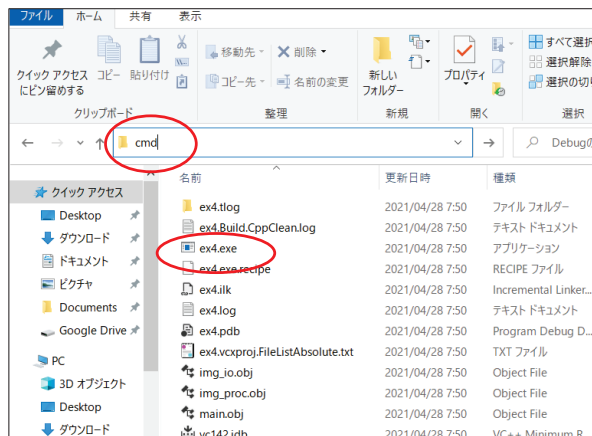
コマンドプロンプト(コマンドライン)

- 文字入力によってプログラムの実行や結果表示を行うもの



コマンドプロンプトの簡単な起動方法(Win)

- 実行したいファイル(exe)があるフォルダを開き、アドレスバーに”cmd”と入力してEnter



【作業課題】

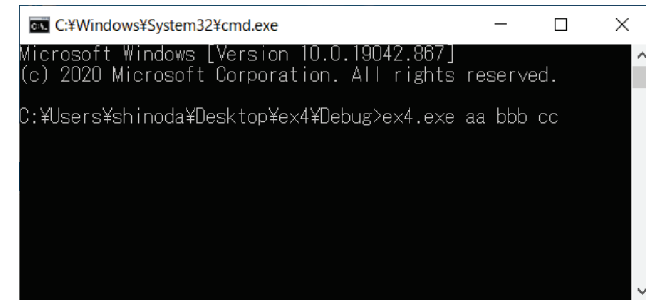
- x64¥Debugフォルダのexeをコマンドライン実行
 - ppmファイルはDebugフォルダ(exeファイルと同一のフォルダ)に移動しておく
 - header.txtを削除しておく
 - Debugフォルダ内を表示した状態で、コマンドプロンプト起動
 - コマンドラインで”ex4.exe”と入力して実行
 - プログラムが実行され、header.txtが作成されることを確認

作業課題の目安: 5分

5

コマンドライン引数

- ex4.exeの後に, 半角空白区切りで文字列を入力
- 空白区切りごとの各文字列(“ex4.exe”, “aa”, “bbb”, “cc”)をプログラム内で使用できる



6

main関数におけるコマンドライン引数

- main関数の引数を下記の通り記載する
 - C言語の規格で引数の型と個数が定まっているため下記以外の書き方はできない

```
int main(int argc, char* argv[])  
{  
    return 0;  
}
```

- プログラム実行時, argcとargvには自動的に値やポインタが割り当てられた状態で開始される

7

main関数におけるコマンドライン引数

- int argc
 - ユーザがコマンドラインで入力した文字列の数
 - “ex4.exe aa bbb cc” の場合, 4
- char* argv[]
 - コマンドラインで入力した各文字列へのポインタ
 - “ex4.exe aa bbb cc”の場合
argv[0]は“ex4.exe”, argv[1]は“aa”, argv[2]は“bbb”...

```
int main(int argc, char* argv[])  
{  
    return 0;  
}
```

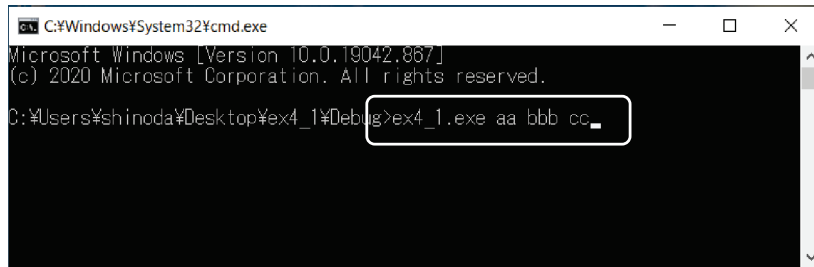
8

コマンドライン引数の挙動例

```
int main(int argc, char* argv[])
{
    int i;
    for (i = 0; i < argc; ++i)
        printf("%s\n", argv[i]);

    return 0;
}
```

このプログラム(ex4_1.c)を作成・ビルドし、
コマンドプロンプトから
> ex4_1.exe aa bbb cc
と実行したとする



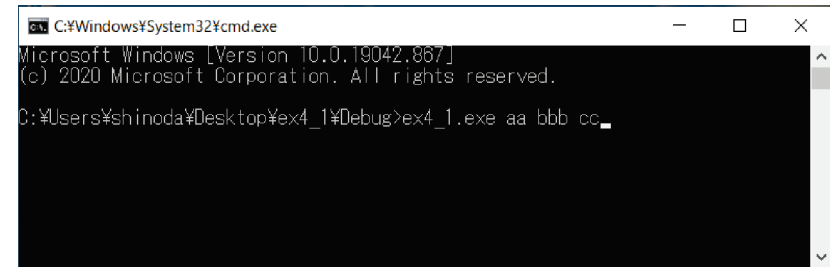
9

コマンドライン引数の挙動例

```
int main(int argc, char* argv[])
{
    int i;
    for (i = 0; i < argc; ++i)
        printf("%s\n", argv[i]);

    return 0;
}
```

} ユーザが入力したコマンドライン引数に従い、
main開始時に自動的にargcとargvに
データが入力されている。
argc ← 4
argv[0] ← "ex4_1.exe"の先頭アドレス
argv[1] ← "aa"の先頭アドレス
argv[2] ← "bbb"の先頭アドレス
argv[3] ← "cc"の先頭アドレス



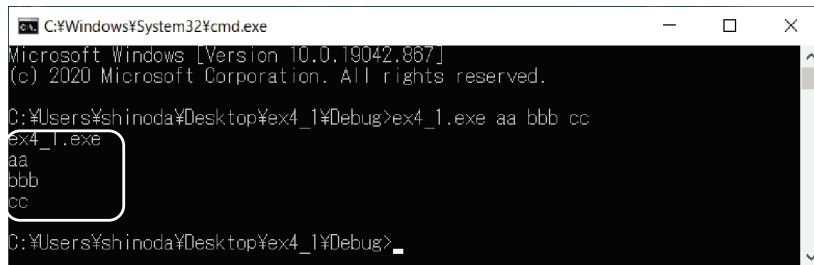
10

コマンドライン引数の挙動例

```
int main(int argc, char* argv[])
{
    int i;
    for (i = 0; i < argc; ++i)
        printf("%s\n", argv[i]);

    return 0;
}
```

} argv[0]: "ex4_1.exe"の先頭アドレスなので、
i = 0では"ex4_1.exe"が出力される。
argv[1]: "aa"の先頭アドレスなので、
i = 1では"aa"が出力される。
.....
つまり、実行すると下記が出力される



11

プログラムの流れ

```
int main(int argc, char* argv[])
{
    IMAGE in_image_data, out_image_data;

    /* 画像情報のロード */
    if (iioLoadFile(&in_image_data, "img01.ppm")) {
        return 1;
    }
    /* 画像情報のコピー */
    ipCopy(&in_image_data, &out_image_data);
    /* 画像情報の書き出し */
    iioSaveFile(&out_image_data, "header.txt");
    /* メモリリークチェック */
    _CrtDumpMemoryLeaks();
    return 0;
}
```

入力ファイル名と出力ファイル名が
固定されているため、
様々な画像を処理したいときに
再ビルドが必要で不便。
scanfで複数回入力させるのも手間。



コマンドライン引数で各ファイル名の
文字列を与えられるように改良する

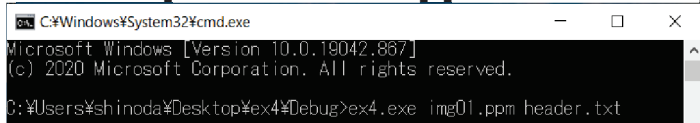
12

【演習課題4_1_2】

- 入力ファイル名と出力ファイル名をコマンドライン引数で与えられるようにmain.cを改良する

- ppmファイルはDebugフォルダ(exeファイルと同一のフォルダ)に移動しておくこと
- コマンドライン入力は下記を想定する

ex4.exe [入力ファイル名] [出力ファイル名]



- コマンドラインから実行し, 出力ファイル名をユーザが自由に変えられることを確認

13

Visual Studioにおける実行時の
コマンドライン引数の設定方法
(以下はデバッグ実行をする場合や
レポート課題では必須)

15

演習課題の目安: 10分

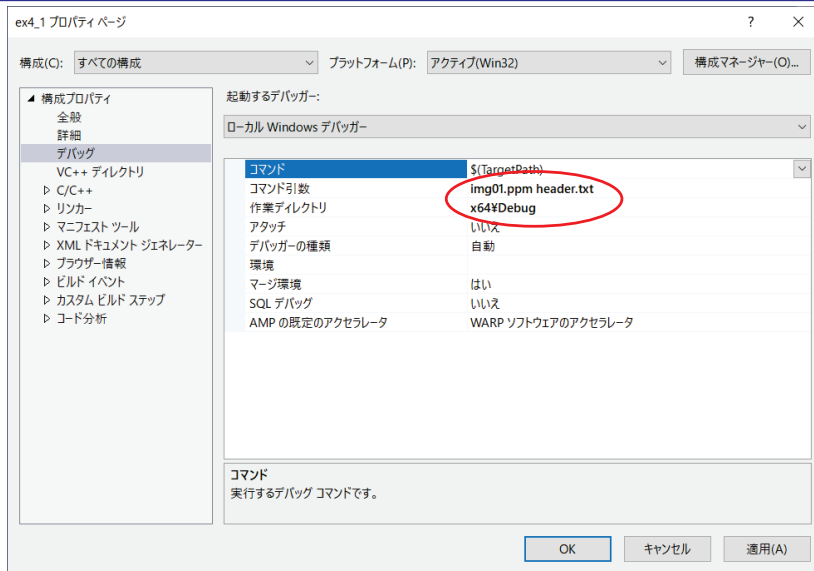
【作業課題】

- Visual Studioのコマンドライン引数の設定方法

- Visual Studioのメニューの「プロジェクト」→「プロパティ」→構成を「すべての構成」→「デバッグ」→「コマンド引数」に, exeファイル名以外の引数を入力
 - img01.ppm header.txt
- また, 同ウィンドウの「作業ディレクトリ」を以下にする
 - x64\Debug
- PPMファイルはDebugフォルダにおいておく
- 演習課題4_1_2と同じくファイルが出力されるか確認
- この方法を使えばコマンドプロンプトの起動は不要

16

設定画面



17

Visual Studio以外の環境の場合

- 各自でデバッグ時のコマンドライン引数の設定方法を調査し、設定すること
 - 設定しないとデバッグ実行ができなくなる
 - メモリリークの確認も困難になる

18