コンピュータ概論　第7回

ユーザインタフェイスと
プログラムへの入出力処理

おさらい
プログラムの作成から実行

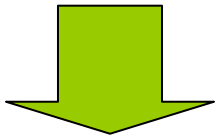
- ソースプログラムの作成
  - エディタなどでプログラミング言語を用いて作成
- コンパイルなど
  - 言語処理プログラムで実行可能なバイナリプログラムに変換
- 実行
  - プログラムとして実行

上記手順は全てコマンドプロンプトを使う

プログラムの実行方法

- Windowsによるプログラムの実行
  - プログラムメニューからの選択
  - プログラムアイコンのダブルクリック
  - 作成したファイルアイコンのダブルクリック
- 昔のコンピュータは？

  マウスが無かった頃はどうしたのか？

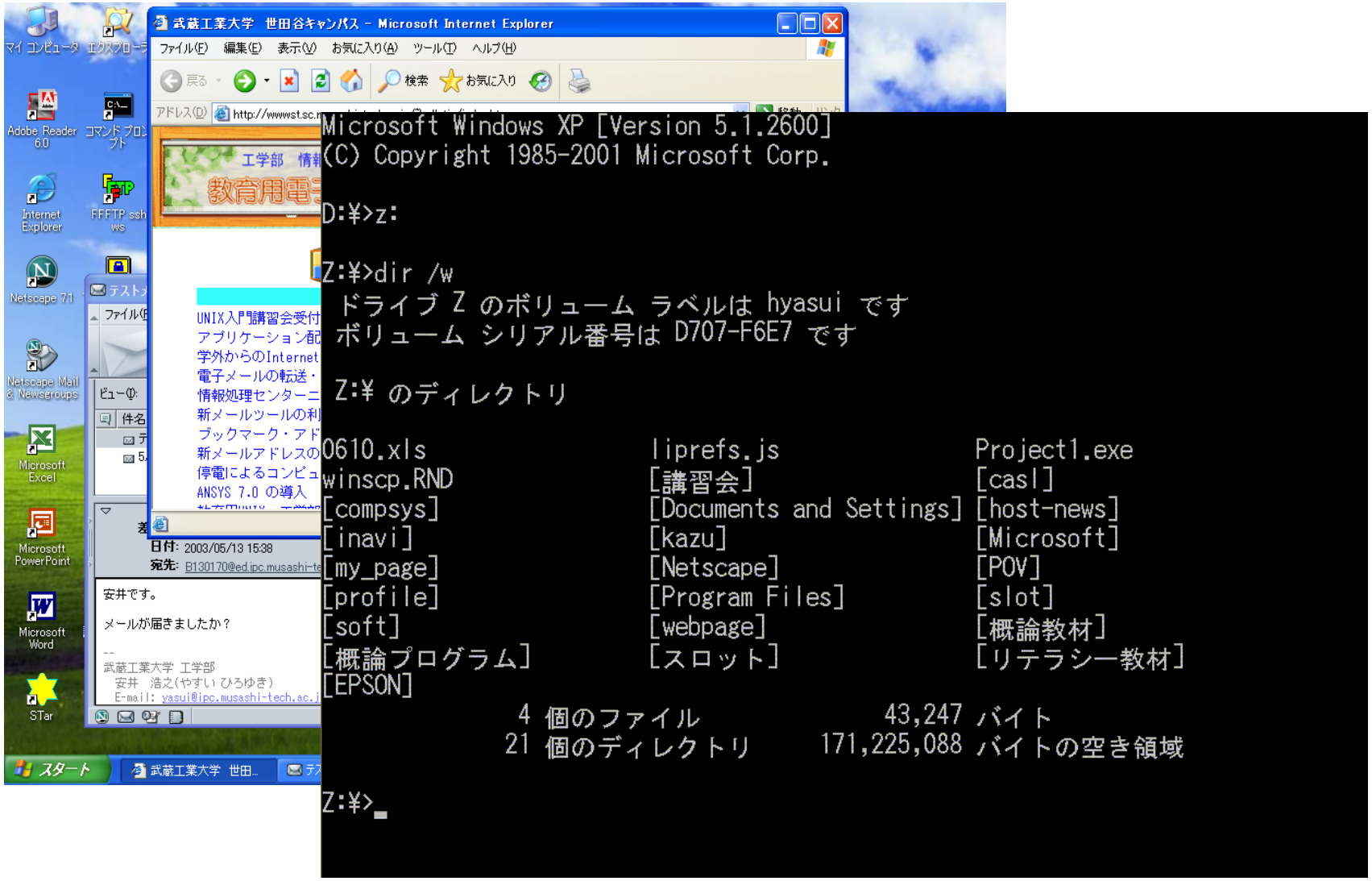


  - 文字を使って命令（コマンド）を入力

ユーザインタフェイス

- 利用者がコンピュータを操作する環境
  - GUI（Graphical User Interface）
    - グラフィカルな表示画面を操作する
    - WindowsやMacOS
  - CUI（Character User Interface）
    - 文字だけの表示画面を操作する
    - UNIXのネットワーク利用（telnet）
    - MS-DOS

GUIとCUI


武蔵工業大学 世田谷キャンパス - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
D:¥>z:
Z:¥>dir /w
ドライブ Z のボリューム ラベルは hyasui です
ボリューム シリアル番号は D707-F6E7 です
Z:¥ のディレクトリ
0610.xls liprefs.js Project1.exe
winscp.RND [講習会] [casl]
[compsys] [Documents and Settings] [host-news]
[inavi] [kazu] [Microsoft]
[my_page] [Netscape] [POV]
[profile] [Program Files] [slot]
[soft] [webpage] [概論教材]
[概論プログラム] [スロット] [リテラシー教材]
[EPSON]
4 個のファイル 43,247 バイト
21 個のディレクトリ 171,225,088 バイトの空き領域
Z:¥>
安井です。
メールが届きましたか？
武蔵工業大学 工学部
安井 浩之(やすい ひろゆき)
E-mail: yasui@ipc.musashi-tech.ac.jp
スタート

WindowsにおけるCUI

- コマンドプロンプト
  - 命令（コマンド）をキーボードで入力して実行するためのウィンドウ（別名：DOSウィンドウ）
- コマンドの例
  - dir　カレントディレクトリ（フォルダ）の中の一覧表示
  - cd　ディレクトリの移動（引数：相対または絶対パス）
  - more　ファイルの内容表示（引数：ファイル）
  - del　ファイルの削除（引数：ファイル）
- 引数（コマンドの後ろに並べて記述）
  - コマンドを実行するときに必要な情報
    - ファイル名やディレクトリ名
    - オプション（コマンドの動作方法を指定）

実習：GUIとCUIの比較

- 同じディレクトリの中をチェックする
  - ネットワークディスク（Z： ）の中を見てみる

Windows（GUI）

マイコンピュータからZ:をダブルクリックで開く

コマンドプロンプト（CUI）

1. コマンド″z:″（Z：への移動，必要なら）
2. コマンド″dir″

プログラムにおけるUI

- プログラムが必要とする情報の入力
  - 引数で指定（通常CUIのみ）
    - 実行開始時に決定（動作中には変更しない）
  - 実行中に入力（GUI・CUI共）
    - 実行途中で入力
- 結果などの出力
  - 画面に文字で出力（CUI）
  - 画面に文字・画像で出力（GUI）

CUIプログラムのUI（入出力）

- 入力
  - プログラム中に入力受付命令を記述
    - scanf　決められた形式で文字を読み取る
    - getc　１文字だけ読み取る
- 出力
  - プログラム中に出力命令を記述
    - printf　決められた形式で文字を表示する
    - putc　１文字だけ表示する

値の入出力

- 入力

  scanfを用いる

- 出力

  printfを用いる

- 変数
  - 入力した値を代入したり，計算結果を保持したりするときに使う

変数

- 数値・文字などの情報を代入できるもの
  - はじめから決まっている値や文字ではないものを扱うときに必要
    - 別の値や文字を代入すると内容を変更できる
  - 変数に対応するメモリ内の記憶領域が存在
    - 変数には代入できる情報の種類を「型」として事前に定義する

変数の型

- 変数へ代入される情報の種類
  - コンピュータはデータの種類によって内部表現（バイナリ表現）が異なる
  - １つの変数はメモリ空間のある領域を確保し、代入された値や文字をその領域に保持する
  - 代入する値は変数の型に一致している必要がある

変数の定義

型名　変数名;

型名　変数名１,変数名２,…;

- 変数名
  - ある特定の語(予約語)を除いた任意の文字列で指定可能(空白や記号文字は原則不可)
- 型名
  - int　　整数
  - char　　文字(半角文字)
  - float　単精度実数(浮動小数)
  - double　倍精度実数(浮動小数)

  など

Cにおける数値表現

- 整数（負数を2の補数で表現）
  - 16ビットまたは32ビット（OSにより異なる）
  - 型の宣言名　`int`
- 実数（IEEE方式の浮動小数）
  - 32ビット（単精度）
    - 型の宣言名　`float`
  - 64ビット（倍精度）
    - 型の宣言名　`double`
  - 整数値について
    - `1.0`や`2.0`のように表記
  - 0について
    - `0.0`と表記

プログラムによる確認

- 型のビット長を調べるプログラム
  sizeof演算子を用いる（バイト数が得られる）

```
#include<stdio.h>
main(void){
   printf("charのビット長は%d¥n",sizeof(char)*8);
   printf("intのビット長は%d¥n",sizeof(int)*8);
   printf("floatのビット長は%d¥n",sizeof(float)*8);
   printf("doubleのビット長は%d¥n",sizeof(double)*8);
}
```

型と演算

- 型変換
  - 他の型に変換する(キャスト演算)
    - intの1をfloatの1.0に

```
float x;
x = (float)1.0;
```

    - floatの1.2をintの1に

```
int y;
y = (int)1.2;
```

型と演算

- 自動型変換
  - 代入先の変数型に変換
    - xは1.0

```
float x;
x = 1;
```

  - 同じ型同士の演算結果はその型のまま
    - intの商は小数点以下切り捨てされる
      xは3（xがfloatなら3.0）

```
x = 10/3;
```

  - 異なる型同士の演算結果は上位型に変換
    - intとfloatならfloatになる
      xは6.9（但しxがintなら6）

```
x = 10-3.1;
```

printf（出力用命令）

- 実行中にデータを表示するための命令
  - システム関数
- 引数（関数を実行するときに必要な情報）
  - 表示する文字列
  - 変数の内容を表示するための文字列と変数
- 実行
  - 指定された文字列を表示
  - 変数がある場合は、その内容を文字列にして表示

printf("表示文字列");

- 表示文字列
  - 表示する文字列
  - センターのシステムでは、日本語も可能
  - 制御文字（cf.ASCIIコード表）も一部使える
    - ¥n　　改行
    - ¥t　　タブ
    - ¥¥　　¥を表示する

printf("表示形式", 変数…)

- 表示形式
  - 以下の形式指定文字を文字列中に入れるとその場所に変数の内容を表示する
  - 対応しない型を指定すると強引にその型として表示
    - %d　整数(int)
    - %f　実数(float)
    - %lf　実数(double)
    - %c　ASCII文字(半角)1文字(char)
    - %s　文字列(1文字以上の文字)
- 変数…(複数の場合は『, 変数, 変数, …』)
  - 中身を表示する変数
  - 形式指定文字の順番に並べる

プログラム例

- 演算結果を表示するプログラム

```
#include<stdio.h>
main(void){
  int x=3,y=4;
  printf("%d+%d=%d¥n",x,y,x+y);
}
```

定義と同時に
代入も可能

変数を含む
数式も表示可能

scanf(入力用命令)

- 実行中にデータを入力するための命令
  - システム関数(システムが用意した命令)
- 引数(関数を実行するときに必要な情報)
  - 入力するデータの型とそれを代入する変数を引数として指定する
- 実行
  - プログラムは入力待ちとなり、入力されるまで次の処理を行なわない
  - 入力待ちであることを特に知らせてくれない
    - 自分で知らせるように作る必要がある

scanf("入力形式", 変数…);

- 入力形式（入力されるデータの型）
  - 以下の形式指定文字を入力する順番に記述
    - %d　整数（int）
    - %f　実数（float）
    - %lf　実数（double）
    - %c　ASCII文字（半角）1文字（char）
    - %s　文字列（1文字以上の文字）
- 変数…（複数の場合は『, 変数, 変数, …』）
  - 入力されたデータの代入先変数
  - 変数名の前に&（アンド）を付ける（文字列を除く）
    - &は、その変数の確保したメモリ空間のアドレス（Cではポインタ）を指し示す記号

プログラム例

- 入力を使うプログラム（三角形の面積）

```
#include<stdio.h>
main(void){
   int a, b, c;
   printf("三角形の面積を求めます¥n");
   printf("底辺＝");
   scanf("%d",&a);
   printf("高さ＝");
   scanf("%d",&b);
   c = a * b / 2;
   printf("底辺%d,高さ%dの面積は%d¥n",a,b,c);
}
```

プログラム例

- 複数の入力を使うプログラム（三角形の面積）

```
#include<stdio.h>
main(void){
  int a, b, c;
  printf("三角形の面積を求めます¥n");
  printf("底辺,高さ＝");
  scanf("%d,%d",&a,&b);
  c = a * b / 2;
  printf("底辺%d,高さ%dの面積は%d¥n",a,b,c);
}
```

プログラム例

- 文字を使うプログラム

```
#include<stdio.h>
main(void){
  char a;
  scanf("%c",&a);
  printf("Hello, %c¥n",a);
}
```

半角文字を入力し
文字型変数aに代入

文字型変数aを表示

プログラム実習

- 入力した半角文字の文字コード(10進)を表示するプログラム(下記)を入力し実行せよ
  - 整数の表示形式(%d)を使うことで強制的に表示できる

```
#include<stdio.h>
main(void){
  char a;
  printf("文字を入力してください¥n");
  scanf("%c",&a);
  printf("%cの文字コードは%dです¥n",a,a);
}
```

おさらい

- ユーザインタフェイス
  - GUIとCUI
- プログラムにおける入出力
  - printf
  - scanf
- 変数
  - 変数型
  - 型の変換

プログラム課題

- 半角文字をscanfで代入した文字型変数に1を足した結果をprintfで文字型で表示するプログラムを作成せよ。
- このプログラムを提出せよ。
- ファイル名：p1.c
- ヒント
  - 代入した変数がxの場合，x＋1を文字型の表示形式で表示するように作る