

生産専攻 2 年前期 画像情報処理工学

第 1 回目の目標 : UNIX と GIMP の操作ができる。プログラムやデータファイル名 pa

1 画像情報処理工学の授業の進め方

- 1) ホームページにシラバスに沿った課題を掲示する
URL: <http://www.ichinoseki.ac.jp/satok/SATOK/ex/index.html>
- 2) 補助資料はインターネット上で検索して利用すること
- 3) 毎回、演習課題に沿って解説し作業する
- 4) 必須課題、応用課題は、番号、氏名、課題名、作成プログラム課題内容、課題プログラムとコメント、必要に応じて作成した画像を貼り付け、特に工夫した部分について簡単な説明文を付けて、印刷し、指定された期限内に提出すること。

2 画像情報処理工学演習のための環境・UNIX 環境の復習と UNIX コマンド

授業でよく使うコマンド

ls	リスト表示	使用例	ls -a , ls -lrt など
cd	ディレクトリ変更	使用例	cd .. , cd tmp など
mkdir	サブディレクトリ作成	使用例	mkdir tmp
cp	コピー	使用例	cp tmp/* prg
mv	移動、ファイル名変更	使用例	mv tmp/* ../
rm	削除	使用例	rm */pgm
od	データ表示	使用例	od -x pa1.pgm more, od -c pa1.pgm more
history	履歴(C シェルの場合)		

ヒストリ機能によるコマンドの簡易入力

- !c 最後に実行した c で始まるコマンドの再実行(再コンパイルするとき)
- !p 最後に実行した p で始まるコマンドの再実行(プログラムの再実行をするとき)
- !! 直前のコマンドの再実行

UNIX コマンド解説サイトの例(検索で使いやすい資料を利用してもらいたい)

<http://www.tj.chiba-u.jp/lecture/prog2/unix.html>
http://www.din.or.jp/~raelian/unix_command.html
<http://www.jsk.t.u-tokyo.ac.jp/~ikuo/softp/unix-commands-contents.html>

3 C プログラムの作成と実行

授業用ウェブからソースコードをコピー&ペーストで行う

エディタによる作成 エディタは専用エディタ(ターミナルソフトや vi など)を使うこと

C コンパイラの仕方

```
cc -o pa1 pa1.c -lm (数式関数を含む場合)
```

実行の仕方

```
pa1 100 100 pa1.pgm
```

左側がコマンド 右側の 100 100 pa1.pgm は、パラメータ

C プログラム(関数)の解説サイト(検索で使いやすい資料を利用してもらいたい)

<http://www9.plala.or.jp/sgwr-t/>

http://cham.ne.jp/piro/p_stdfunc.html

<http://www5c.biglobe.ne.jp/~ecb/c/c00.html>

4 ラスター画像表示・レタッチソフト GIMP とその使い方

GIMP:ラスター画像データ表示/加工用のフリーソフト

JPEG や PNM など、各種画像保存形式への対応可能

本授業ではPNM形式のうち、pgm(グレー画像)、ppm(インターレースカラー画像)を使い、その表示目的とする。

また JPEG 画像の pgm, ppm 画像への変換ツールとして使う。

GIMP は photoshop に相当する高度な機能が付加されている。これはフリーソフトで、Windows 用もある。

GIMP は本授業では表示目的、画素の調査、保存形式変更目的だけしか使わない。

これ以外の機能は、後で示すサイトで調べてもらいたい。

4-1: 本授業で使う PGM 形式の画像データ例(グレースケール画像。640x400 サイズで 1 画素8ビットの場合)

P5 ←— 最初に P5 の文字キャラクタがありグレー画像の raw 画像であることを示す

INCT ←— コメント(何文字でも可)

640 400 ←— 画像サイズの文字 横(ピクセル)が 640、縦(ライン)400 の画像データ

255 ←— 整数最大値が 255 であり、8 ビット画像データである

(この後に 8 ビット画像データが 640x400 バイト続く。

例えば最初のデータが 255 (16 進数で FF)なら白を示す。)

4-2: 本授業で使う PPM 形式の画像データ(カラー・インターレース画像。640x400 サイズで 1 画素8ビットの場合)

P6 ←— 最初に P6 の文字キャラクタがありカラー画像の raw 画像であることを示す

INCT ←— コメント(何文字でも可)

640 400 ←— 画像サイズの文字 横(ピクセル)が 640、縦(ライン)400 の画像データ

255 ←— 整数最大値が 255 であり、8 ビット画像データである

(この後に 8 ビットカラーのインターレース画像データが 640x400x3 バイト続く。

最初が 255 0 0 なら、赤を示す。インターレースの並びは RGB の順となる)

png 画像(pgm,ppm)の保存形式の解説はこちら

<http://www.mm2d.net/c/c-08.shtml>

<http://www.aso.ecei.tohoku.ac.jp/~kato/gzipped-pnm/gzipped-netpbm-ja.html>

<http://www.media.eng.hokudai.ac.jp/~chicchi/H14/exam1/chapter05.html>

GIMP ソフトの解説はこちら

<http://ja.wikipedia.org/wiki/GIMP>

<http://hp.vector.co.jp/authors/VA025935/gimp-help/gimp/contents.html>

5 モノクロ画像を作成するサンプルプログラム pa1.c

このプログラムをコピーして一部改良し、改良後は課題番号に従って pa2.c,pa3.c としてファイル保存。
画像ファイルも、この課題のものであることがわかるように pa1.pgm というように、プログラムに対応した名称とする。これらは後で、USB メモリ等でファイル提出を依頼することがある。

以下のものは、画素値 128 が一様に並ぶ画像を、指定されたサイズで生成するものである。

なお画素値=0 は黒、255 は白である。

サイズはプログラムを実行する場合に指定し、例えば

```
pa1 300 200 pa1.pgm
```

とすると、コンパイルしたモジュール pa1 を実行し、横 300 画素、縦 200 画素の画像ファイル pa1.pgm が作成される。ファイルのバイト数は ls -l で確認できるが、pnm 形式ヘッダに 60000 バイトの画像生データの値になる。

プログラムはウェブからコピー&ペーストでエディタへ貼り付ける。エディタは使いやすいものを使ってかまわない。説明は UNIX で進めるが、ワープロ作成作業等で必要なら windows を併用してもかまわない。

```
/* -----  
Image Processing pa1.c  
PGM Format Gray Image Data Generation  
Ichinoseki National College Advanced Course  
----- */  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
  
main(argc,argv)  
int argc;  
char *argv[];  
{  
    static int i,j,m;  
    static int pixel,line;  
    FILE *fp1;  
  
    /* Command parameter check */  
    if (argc!=4){  
        printf("Usage : command pixel line output_file¥n");  
        return 1;  
    }  
  
    /* Image size parameter load */  
    sscanf(argv[1],"%d",&pixel);  
    sscanf(argv[2],"%d",&line);  
  
    if ( pixel*line>1000*1000){  
        printf("Over image size¥n");  
        return 1;  
    }  
}
```

```

/* Output image file name read */
if ((fp1=fopen(argv[3],"w"))==NULL){
    printf("Can't open output_file¥n");
    return 1;
}

/* pnm(pgm) header code output */
fprintf(fp1,"P5¥n"); /* Gray binary image data */
fprintf(fp1,"# INCT¥n"); /* Comment */
fprintf(fp1,"%s %s¥n",argv[1],argv[2]); /* Pixel Line Image size */
fprintf(fp1,"255¥n"); /* Image data maximum value */

/* Gray binary image data output */
for (i=0; i<line; i++)
{
    for (j=0; j<pixel; j++)
    {
        m=128;
        putc(m,fp1);
    }
}
fclose(fp1);
return 0;
}

```

6 画像データを作成し、ファイルを観察してみる

1) サンプルプログラムによる画像ファイルの生成

```
pa1 200 200 pa1.pgm
```

8ビット、200x200サイズのグレー画像 pa1.pgm を作成する。生画像データのサイズは 40000 バイト。

2) データサイズはいくらか？

ls -l コマンドで pa1.pgm ファイルのサイズを確認してみよう。何バイトであったか？

40000 を超えたバイト数が、上のリストの /* pnm(pgm) header code output */ 部分で追加した画像形式を示す目的のヘッダー情報の容量である。

3) 画像データの中身の表示(od コマンド)

od -x pa1.pgm | more で 16 進数の画像データファイルを表示

od -c pa1.pgm | more で文字による画像データファイルを表示

3) GIMP による表示観察

画像ファイルを「開く」で pa1.pgm ファイルを探し、表示してみよう

授業では pgm 形式のまま扱うが、必要に応じて jpeg 画像等に形式変更して保存できる。

4) 画像ファイルサイズを変更し、いろいろな画像を作成してみる

```
pa1 100 100 pa1.pgm 100x100 サイズの画像になる
```

```
pa1 640 400 pa1.pgm 640x400 サイズの画像になる
```

7 本日のプログラムの改良課題(必須)

残す時間を、上のサンプルプログラムを改良して、次のプログラム及び画像ファイルを作りなさい。

(1) 画素の値が 200 で、画像サイズが 300x300 のグレー画像 pa12.pgm という画像ファイルを作成する。
プログラムファイルは pa12.c とする。

(2) 白地の中央に縦方向に黒線を引いた 300x300 サイズの画像を作る。
プログラム名を pa13.c とし、画像ファイル名を pa13.pgm とする。

ヒント： 横方向 j の位置 50 のところに黒($m=0$)を置くプログラムの例

1) if 文を使う場合

```
m=255;
if (j==50){
    m=0;
}
putc(m,fp1);
```

2) 判定文を使う場合

```
m=(j==50)? 0:255;
putc(m,fp1);
```

と書くと、上の結果となる。なぜこうなるか、コメント等を付けて、指定した期限までに報告すること。

8 プログラムの応用課題

これは自学自習課題である。

画素値 0 である画像サイズ 200x200 の画像データに、縦と横それぞれ 20 画素おきに白い罫線が引かれている画像 pa14.pgm を作成したい。そのプログラム pa14.c とコマンドラインおよび画像を示し、コメントを書き添えて提出しなさい。

この応用課題は、期末試験の問題のひとつと考えている。

グループで開発してもよいが、どんな原理で画像作成しているか、各自、なぜそうなるか、理由がわかっていることが大事である。