第３回ＣＡＤセミナー

主催：日本放射線技術学会　画像分科会
共催：学術委員会，中部部会

日時：2001年2月3日　午後１時より５時まで
場所：岐阜大学工学部応用情報学科／会議室および演習室

＜プログラム＞

12:30　　受付開始（応用情報学科１階会議室C127）

1:00 - 1:10　挨拶，概要説明（応用情報学科１階会議室C127）
岐阜大学工学部：藤田広志（画像分科会・分科会長），原　武史（分科会委員）

1:10 - 1:50　教育講演（応用情報学科１階会議室C127）
金沢大学医学部：真田　茂（画像分科会委員）
内容：画像処理プログラミングについての教育講演

2:00 - 4:30　演習（応用情報学科７階会議室C728, 730）

4:30 - 5:00　研究室，学内見学
工学部応用情報学科・藤田研究室の見学，他

その後，岐阜市内において懇親会を行います．
岐阜大学バス停から17:30に会場行きのマイクロバスが出ます．

＜セミナースタッフ＞

画像分科会　：藤田広志（岐阜大学），真田　茂（金沢大学），原　武史（岐阜大学），市川勝弘（名古屋市立大学）
中部部会　　：津坂昌利（名古屋大学）
補助スタッフ：福岡大輔（岐阜工業高等専門学校），松原友子（名古屋文理大学），李　鎔範（新潟大学）
畑中裕司（岐大D2），中川俊明（岐大D2），篠原範充（岐大D1），牛場洋明（岐大D1）

この資料は，第３回ＣＡＤセミナーにおいて配付したものです（2/3/2001作成）．
また，セミナーで用いた画像の一部は日本放射線技術学会発刊の標準ディジタル画像データベース
（胸部腫瘤陰影像および非腫瘤陰影像）を利用しています．

予備演習：Ｃ言語によるプログラミングの基礎

こちら側には作業を書きます	こちら側にはLINUX上での入力を主に書きます
ログイン後， **cad-semというディレクトリに移動し**	% cd☆cad-sem●　：☆はスペース ：●はリターン ：%は入力不要
さらにprogramsというディレクトリに移動， **さらにYOBI　というディレクトリに移動します．**	% cd☆programs● % cd☆YOBI●　：大文字小文字は区別されます
あと，もう１枚，端末ウィンドウ（kterm）を開きましょう．	ktermのボタンをおすと開きます.
＜実験0-1：ファイル名を見る＞ そこにあるディレクトリを見てみましょう． １から６までの番号の付いたディレクトリは見れましたか？	% ls●
＜実験0-2：ディレクトリをかわる＞ 1.hello_CADに移動してみましょう． そのディレクトリにあるファイルを見てみましょう.	% cd☆1.hello_CAD● % ls●
＜実験0-3：ともかくコンパイルする＞ このディレクトリにはプログラムがあります．これをコンパイルしてみましょう． プログラムは，main.cというファイルに書かれています.	% gcc☆main.c●
＜実験0-4：ともかく実行する＞ 次は，実行してみましょう． hello CAD worldsと表示されましたか？	% a.out●
＜実験0-5：プログラムの中を見る＞ 次は，表示されるメッセージを自分の好きなメッセージに変えてみましょう．そのためには，プログラムを変更する必要があります． プログラム（main.c）はテキストファイルで書かれていますので，そのテキストを修正し	

ます．そのためには，テキストエディタを利用します．LINUXでは，mule（みゅーる）というテキストエディタが使われます．muleを使ってmain.cを開いて修正できるようにしましょう．
ウィンドウが開いて中にプログラムが表示されましたか？

% mule☆main.c●

＜実験0-6：プログラムを修正する＞
今度は，そのプログラムの中身を修正して，メッセージを書き換えましょう．
printf("hello CAD worlds");
と書かれた行を探して（下の方），その中を書き換えて下さい．
そして，そのファイルを保存して下さい．

カーソルの移動は矢印キーでできます．

ファイルの保存は，
「コントロールキー」を押しながら，
「X」，次に「S」を押します．

＜実験0-7：再びコンパイル＞
新しくファイルが修正できたので，コンパイルして実行してみましょう．
方法は，実験0-3と実験0-4の繰り返しです．

作業のコツ
コンパイル用のウィンドウ，実行用のウィンドウを別に開いておくと作業が分かり易く，はかどります．

＜実験0-8：再び修正＞
再び，メッセージを変えてみましょう．そして，くり返し行い，ファイルの修正，コンパイル，実行，という手順をマスターしましょう．

端末ウィンドウでコントロールキーを押しながら「p」を押すと直前のコマンドが表示され，リターンキーを押すとそれを実行できます．

今度は，ディレクトリ2.hello_CAD2へ移動して下さい．今は，1.hello...にいますので，一度，１つ上に移動して
そこで，2.hello...へ移動します．

% cd☆../●
% cd☆2.hello_CAD2●

＜実験0-9：２つめのメッセージ表示＞
まえのプログラムでは，メッセージを変更するためには，プログラムを修正して，コンパイルしなおす必要がありました．今度は入力されたメッセージを出力するプログラムに改造します．まず，コンパイルしてみましょう．
そして，実行してみましょう．
こんどは，実行するときに表示したいメッセ

（☆と●はここでおしまいです）

% gcc main.c

ージも入力します．表示したいメッセージを
abcdefgh
として，実行しましょう．
結果はどうなりましたか？

% a.out abcdefgh

First arg(#0)にはa.out
Second arg(#1)には入力したメッセージが表示されたと思います．

このように，プログラムに入力を与えることができます．そして，特にこのように実行するコマンド（この場合はa.out）につけて入力を与えることを「コマンドラインで入力を与える」といいます．非常に重要なテクニックです．

今度は，足し算をするプログラムを作りましょう．ディレクトリ3.for_and_printfへ移って下さい．

（がんばって移動しましょう）

＜実験0-10：足し算をするプログラム＞
コンパイル
実行
を試して下さい．

% gcc main.c
% a.out

segmentation faultと表示されましたか？
これは，プログラムの実行中にエラーが発生したことを示しています．
実は，このプログラムもコマンドラインから数字を入力する仕様になっています．
再び実行してみましょう．
何が表示されましたか？
SUM = 55となりましたか？
このプログラムは，コマンドラインから２つの数字（引き数「ひきすう：といいます）をとり，その間にある整数の和を計算します．

% a.out 1 10

＜実験0-11：中を見る＞
muleで中を読んで，コメントなど読んで下さい．for文，printf文の使い方が分かると思い

% mule main.c

ます.

今度は足し算する数字に条件をつけましょう．このように条件をつける場合にはifという文章を利用します．
4.if_and_forへ移動しましょう．

（がんばって移動しましょう）

＜実験0-12：コンパイル，実行，中を見る＞
コンパイル
実行
中を見る
で同じように作業して下さい.
いろいろな値で試してみて下さい.
元の状態では，0から1000までの数しか足さないようになっています.

% gcc main.c
% a.out 1 10
% mule main.c

% a.out -10 2000はどうですか？

＜実験0-13：最大値，最小値の設定を変更＞
muleでプログラム（main.c：ソースファイルともいいます）を開いて，設定している部分を変更しましょう.
そして，再び
コンパイル,
実行,
で試してみましょう.

% gcc main.c
% a.out 好きな値　好きな値

プログラミング言語で数字を表すのに決まりがあります．大きくは，実数と整数です．細かくは，その数字を表すのに必要なビット数，ということになります．ここでは，その数を調べてみます．結構，重要です．

＜実験0-14：変数のいろいろ＞
5.size_ofというディレクトリに移動して,
コンパイル
実行
を試してみた後,
muleでmain.cを開き,
main.cのコメントをよく読んで下さい.

（一生懸命移動しましょう）
% gcc main.c
% a.out

% mule main.c

＜実験0-15：変数のいろいろ(2)＞
バイト数は確認できましたか？
（これはけっこう重要）
一見簡単な割算も，計算機では微妙な問題になりそうなのが確認できましたか？

予備演習の最後に，画像処理でもっともよく使われる保存形式の配列について実験します.
6.arrayに移動して下さい.　（がんばって）

＜実験0-15：配列の準備＞
コンパイル　% gcc main.c
実行　% a.out
を試してみた後,
muleでmain.cを開き,　% mule main.c
main.cのコメントをよく読んで下さい.
ここでは，配列について，詳しく説明しています.

<Check yourself>

コンパイルはできましたか？

Linuxでは，gccというC/C++コンパイラを利用しています.

プログラムの中身は見れましたか？

Linuxでは，muleというテキストエディタ（ワープロみたいなもの）を使います.

コマンドラインからの文字，数字の入力はできましたか？

main()という関数がその数字を受けます．atoiで文字列を整数に変換します.

for文は使えましたか？　if文は使えましたか？

くり返し同じ処理を行うのはfor文で実現します.
大小判定．同値判定などがif文で可能です．詳しくはC言語の本を参考に.

変数のバイト数は理解できましたか？

１画素が１バイト，２バイトはここできいてきます.

配列は使えそうですか？

大きな画像, 小さな画像も１つで扱うには6.arrayのように利用するのがお勧めです.

演習１：画像の入出力と表示

この演習はprogramディレクトリのno.1というディレクトリの中で行います．

演習の第１段階は，画像ファイルをコピーするプログラムを作ります．何も処理をしません．しかし，ここには，いくつかの重要な手順が含まれています．

C言語では，処理全体を細かい「関数」の集まりとして（全体も関数）扱います．データは関数へ渡して，その関数で処理を行います．このことは，プログラムを考える上で非常に重要です．

演習１のサンプルプログラムは，そのプログラムになります．ここでは，まず，ハードディスク上に保存された画像ファイルを開き，それをコンピュータのメモリ上に保存します．そして，そのメモリ上の画像をハードディスク上に別名をつけて保存します．

それぞれが，関数で表現されており，main文のなかから，いろいろな関数が 順序よく呼ばれて，処理が行われます．

＜実験1-1：コンパイル＞

プログラムをコンパイルしてみましょう．いままでは，すべて実行ファイルがa.outでしたが，「-o」オプションで自由な名前をつけられます．proc1という実行ファイルで作りましょう．

（No.1ディレクトリに移動した後）

```
% gcc -o proc1 main.c
```

注意：-oは「オー」です．

＜実験1-2：実行＞

実行してみましょう．ここでは，コマンドラインから「入力画像のファイル名」，「画像の幅」，「高さ」，「出力画像のファイル名」を指定します．

入力画像名：chest.raw
幅，高さ：それぞれ512
出力画像名：好きな名前

これで，入力画像をコンピュータのメモリに保存し，その内容を再びディスクに書き出す，という事ができました．

```
% proc1 chest.raw 512 512 out.raw
```

＜実験1-3：ソースファイルを見る＞

muleでソースファイルを読んでください．コメントなどを確認すると，動作の流れが分かります．

```
% mule main.c
```

＜実験1-4：画像表示＞

表示用のプログラムがあるので，画像のファイル名，画素の幅，高さを入力します．

```
% x8view ファイル名　幅　高さ
% x8view chest.raw 512 512　：原画像表示
% x8view chest.raw 512 512　：処理画像表示
```

演習２：白黒反転

この演習はprogramディレクトリのno.2というディレクトリの中で行います．

演習の第２段階は，画像の白黒を反転するプログラムを作ります．同時に画像全体の画素値の平均を求めるプログラムも作成します．ここでは，

・白黒を反転する関数
・画像の画素値の平均値を求める関数

を作成して，そこへ元画像を保存した配列，反転画像の配列などを渡し， 処理結果を得ることとします．

演習２のサンプルプログラムは，そのプログラムになります．ここでは，まず，ハードディスク上に保存された画像ファイルを開き，それをコンピュータのメモリ上に保存します．演習１とは異なり，元画像用の配列，反転画像用の配列の２つが宣言されていることに注意して下さい．

＜実験2-1：コンパイル＞

演習１と同じようにコンパイルしましょう．実行ファイル名はprocにしましょう．

（no.2ディレクトリに移動した後）
% gcc -o proc main.c

＜実験2-2：実行＞

実行してみましょう．実行方法も演習１と同じです．そして，表示しましょう．
　入力画像名：chest.raw
　幅，高さ：それぞれ512
　出力画像名：好きな名前

% x8view chest.raw 512 512 ：原画像表示
% proc chest.raw 512 512 chest.raw.inv
% x8view chest.raw.inv 512 512

＜実験2-3：ソースファイルを見る＞

muleでソースファイルを読んでください．コメントなどを確認すると，動作の流れが分かります．

% mule main.c

＜実験2-4：反転の反転を行うように改造＞

画像を白黒反転した後に，再び白黒反転するように改造しましょう．muleでmain.cを開き，main関数の中に反転を行っている関数を追加します．そして，その画像についても画素値の平均値を算出してみましょう．使い方は，ソース，コメントを参考にして下さい．

反転を行う関数：
　invert_image(..)
画像の平均値を求める関数：
　get_average(...);

演習３：階調処理

この演習はprogramディレクトリのno.3, no.3-1, no.3-2のディレクトリの中で行います．

演習の第２段階は，画像の白黒を反転するプログラムでしたが， こんどは画像のコントラストを変換する手法を取り扱います． 一般に医用画像は，コンピュータが表示できるよりも多くの情報量を持っています．ＣＴ画像においてはCT値を表示用の濃淡に変換するために，ウィンドウ処理を 行いますが，これはまさに階調処理と同じです．ここでは，

・8bit画像の階調処理（no.3）
・符号無し１０ビット画像の処理：ディジタル化された胸部Ｘ線写真の処理（no.3-2）
・CT画像の処理：自由なウィンドウレベルとウィンドウ幅での処理（no.3-3）

を行うプログラムを作成し処理結果の８ビット画像を得ることを目的とします．

＜実験3-1：コンパイル＞

同じようにコンパイルしてみましょう．ただし，今度はソースファイルが２つに分割されているので，注意が必要です．

（no.3ディレクトリに移動した後）
% gcc -o proc main.c winproc.c

＜実験3-2：実行＞

同じように実行してみましょう．レベル，幅が尋ねられます．そして，表示しましょう．

% proc chest.raw 512 512 out.raw
幅とレベルの入力が促される．
% x8view chest.raw.inv 512 512

＜実験3-3：ソースファイルを見る＞

muleを使ってファイルの中身を見ましょう．

% mule main.c，% mule winproc.c

no.3-2に移動して同じように実験します．

＜実験3-4：Makefileを使ってコンパイル＞

Makefileとは，コンパイルの条件を書いた設定ファイルです．この場合，makeと打つとコンパイルできます．そして，実行しましょう．

% make
% proc chest.raw 1024 1024 out.raw
　　この画像は10bit画像です．
% x8view out.raw 1024 1024

＜実験3-5：プログラムの改造＞

表示したら白黒反転していませんか？正しく反転した状態で表示されるように，プログラムを改造しましょう．

% mule main.c　　：プログラムの編集
% make　　：コンパイル
% x8view ファイル名　1024 1024
　　：画像表示

no.3-3に移動して，同様に実験を行います．

＜実験3-5：CT画像の表示＞

コンパイルして，実行，表示を試みましょう．

（省略）
CT画像はshort型で読まれいることに注意して下さい．

演習４：空間領域でのフィルタリング

この演習はprogramディレクトリのno.4, no.4-2, no.4-3のディレクトリの中で行います．

空間領域では，画像への畳み込み演算という手法でいくつかのフィルタ処理が可能になります．ここでは，その代表的なフィルタを２つ取り扱い，また，それらフィルタ処理と組み合わせることによって 実現できるアンシャープマスクフィルタを作成します．以下の３つの演習からなります．

・３×３の平滑化フィルタ（no.4）
・３×３のラプラシアンと原画像に加算して画像をシャープに（no.4-2）
・３×３の平滑化フィルタを利用したアンシャープマスクフィルタ（no.4-3）

以上を行うプログラムを作成し，処理結果の８ビット画像を得ることを目的とします．なお，コンパイルするためには，Makefileがすべて用意してありますので，makeを入力するだけで可能です．

＜実験4-1：平滑化フィルタ＞

no.4のディレクトリに移動後,
コンパイル,
実行,
表示を行って下さい．
また，convolve.cの中の重み係数を変更してみて下さい．さらに可能な方は，5x5の平均をとるように変更して下さい．

```
% cd no.4
% make
% proc chest.raw 512 512 out.raw
% x8view out.raw 512 512
% mule convolve.c
```

足し算，かけ算の項が25になります．

＜実験4-2：ラプラシアンフィルタ＞

no.4-2のディレクトリに移動後,
コンパイル,
実行,
表示を行って下さい．
また，convolve.cの中の重み係数を変更してみて下さい．さらに可能な方は，5x5の領域で，自由なラプラシアン風フィルタを作成ください．

```
% cd no.4-2
% make
% proc chest.raw 512 512 out.raw
% x8view out.raw 512 512
```

やはり，足し算，かけ算の項が25になります．

＜実験4-3：アンシャープマスクフィルタ＞

no.4-3のディレクトリに移動後,
コンパイル,
実行,
表示を行って下さい．強調係数が尋ねられます．鮮鋭成分のみの画像を作成し，階調処理を行い，表示して下さい．ソースファイルを見て，流れを理解して下さい．

```
% cd no.4-3
% make
% proc chest.raw 512 512 out.raw
% x8view out.raw 512 512

% mule main.c
```

演習５：周波数領域でのフィルタリング

この演習はprogramディレクトリのno.5, no.5-2というディレクトリの中で行います．

空間周波数領域でのフィルタリングを取り扱います．

空間領域では，畳み込み演算によって実現していたフィルタ処理が，空間周波数領域で行えるようになります． ここでは，まず画像を空間領域（画素値の領域）から空間周波数領域（スペクトルの領域）へ変換する必要があります．その方法は，フーリエ変換というものです．したがって，空間周波数領域での処理の流れは以下のようになります．

1.画像の読み込み：ハードディスクからchar型の配列へ．
2.画像のフーリエ変換：char型をdouble型に変換．
3.スペクトルに対する処理：スペクトルは複素数で表される
4.スペクトルの逆変換：周波数領域から空間領域へ
5.画像保存

演習は，以下の２つを行います．

・画像のフーリエ変換，逆変換 （変換してすぐに戻すので，画像は変わりません）
・スペクトルへのフィルタ処理による高周波遮断フィルタ

以上を行うプログラムを作成し，処理結果の８ビット画像を得ることを目的とします．

＜実験5-1：画像のフーリエ変換，逆変換＞

no.5のディレクトリに移動後,
```
% cd no.5
```
コンパイル,
```
% make
```
実行,
```
% proc chest.raw 512 512 out.raw
```
表示，を行って下さい．
```
% x8view out.raw 512 512
```
また，main.cの中身を見て,

入力画像の保存配列	*in, mallocで確保されています．
フーリエ変換用の配列（複素数）	*redata, *imdata． これもmallocで．
フーリエ変換の関数への配列の渡し方	FFT2がフーリエ変換の関数です．
フーリエ逆変換の関数への配列の渡し方	FFT2の引き数を変えると逆変換になる．
出力用画像への変換関数	char2double(..), double2char(..)が変換．

を見つけ，その中身も見て下さい．

＜実験5-2：高周波遮断フィルタ＞

no.5-2のディレクトリに移動後,
```
% cd no.5-2
```
コンパイル,
```
% make
```
実行,
```
% proc chest.raw 512 512 out.raw
```
表示，を行って下さい．途中で尋ねられるのは，遮断する高周波成分の量です．
```
% x8view out.raw 512 512
```
500位を指定して下さい．

＜実験5-3：低周波遮断フィルタへの拡張＞

filter.cの中のfreq_filtering関数を改造して，低周波領域を遮断するようにしましょう．

遮断する領域を0.0でうめれば実現できます．実数，虚数とも忘れずに．

＜参考文献＞

Ｃ言語関連：

たくさんあります．どれも同じようですが，以下は，初級以上にお勧めです．

Ｃ言語ポインタ完全制覇　前橋和弥　著　技術評論社　ISBN4-7741-1142-2　2280円
　初級以上の人がつまずく点をよくまとめてあると思います．
　このほか，アルゴリズム辞典風のものが１冊あるとよいでしょう．

Ｃ言語による画像処理関連：

わりとよさそうなものを２冊．

Ｃ言語による画像処理入門　安居院　猛，他，著　昭晃堂　ISBN4-7856-3124-4 3000円
　パタン認識，ニューロ，GAについても記述がある，きっと数少ない本．

Ｃ言語で学ぶ実践画像処理　八木伸行，他，著　オーム社
Ｃ言語で学ぶ実践ディジタル映像処理　八木伸行，他，著　オーム社
　きっとどこの研究室にいってもある本です．
　２次元配列を使ってすべて記述しているので，他への応用を考えた場合には，
　プログラムを書き直す必要があります（そんなに難しいことではありませんが）．

＜付録＞

配付したCD-ROMのprogramsディレクトリ中にあるx8viewディレクトリには，演習で用いた画像表示プログラムが保存されています．Makefileも一緒に保存してありますので，コンパイルして利用ください．作成者は，岐阜工業高等専門学校／福岡大輔先生です．この範囲での利用は快諾をいただいています．

この表示プログラムには，OpenGLという標準的なグラフィック関数とGLUTというその拡張セットを利用していますので，ひょっとしたらWindows系でもそのままコンパイルできるかもしれません．

Linuxをインストールする場合には，必ずグラフィック環境も一緒に入れておいて下さい．Linuxでは，OpenGLはサポートされておらず，その代わりにMesaGLというOpenGL互換のGL環境が与えられています．GLUTについても同様です．

コンパイル方法　：make
実行方法　　：x8view　表示画像ファイル名　幅（画素数）　高さ（画素数）

この冊子は，岐阜大学工学部　原　武史が作成しました．