

前期中間試験 (5E 計算機応用)

2006 年 6 月 11 日

1 UNIX の使い方

1.1 ファイルとディレクトリー

複数のディレクトリーが図 1 のようなツリー構造で管理されている。各ディレクトリーには、複数のファイルやディレクトリーが存在する。図中の * で示した D4 をカレントディレクトリーとする。以下の設問に答えよ。

- [問 1] 矢印に示したディレクトリー D6 内のファイル f1 を指定したい。それを、絶対パスで書け。
- [問 2] 前問の f1 を相対パスで書け。
- [問 3] カレントディレクトリーとは何か?—説明せよ
- [問 4] 親ディレクトリーとは何か?—説明せよ
- [問 5] サブディレクトリーとは何か?—説明せよ
- [問 6] カレントディレクトリーを表示させるコマンドを示せ。
- [問 7] カレントディレクトリーにあるファイルやサブディレクトリーを表示させるコマンドを示せ。
- [問 8] 親ディレクトリーに移るコマンドを示せ。
- [問 9] サブディレクトリーに移る方法を示せ。コマンドとその書き方を示すこと。
- [問 10] ホームディレクトリーに移るコマンドを示せ。
- [問 11] ファイルとディレクトリーの違いについて、説明せよ。
- [問 12] 図 2 のように変更するコマンドを示せ。

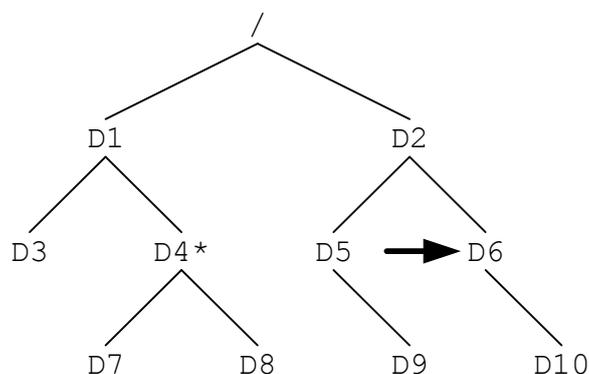


図 1: ディレクトリーのツリー構造

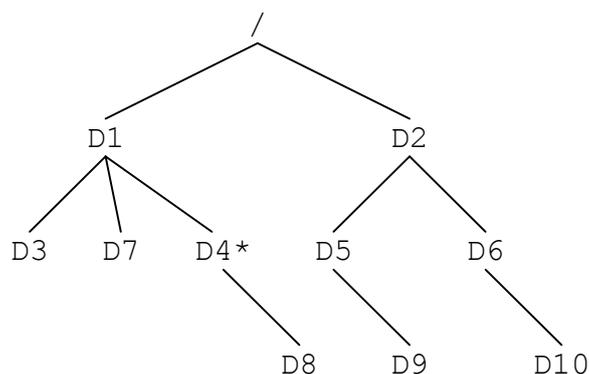


図 2: 変更後のディレクトリー

1.2 コマンド

以下の作業を行う場合のコマンドを示せ。

- [問 1] C 言語のソースファイルに、必要な拡張子を示せ。
- [問 2] program.c という名前のソースファイルを編集したい。ターミナル上でのコマンドを示せ。

- [問 3] カレントディレクトリーにある program.c をコンパイルして, hoge という実行ファイルを作成するコマンドを示せ. ただし, program.c には数学関数が含まれていないものとする.
- [問 4] 前問の program.c に数学関数が含まれている場合のコンパイル方法を示せ.
- [問 5] カレントディレクトリーに hoge という実行ファイルがある. 実行方法を示せ.
- [問 6] ヒストリー機能を説明せよ. そして, その機能を使う方法を示せ.

2 C 言語の文法

2.1 基礎

- [問 1] 通常の C 言語のプログラムは, 多くの関数から構成される. 真っ先に実行される関数を示せ.
- [問 2] 整数型の変数 i と j, fuga を使いたい. その宣言文を書け.
- [問 3] 倍精度実数型の変数 x と y, hoge を使いたい. その宣言文を書け.
- [問 4] 倍精度実数型の変数を 10000 個使いたい. どのようにするか?. 宣言文を書け. 変数名等は回答者が適当に決めよ.
- [問 5] 整数型の変数の値が次のとおりだったとする.

```
a=10    b=3    c=0
```

以下の演算結果を示せ.

```
(a) a%b    (b) a==b    (c) a!=b
(d) !c     (e) b||c    (f) b&& c
```

- [問 6] 整数型の変数の値が次のとおりだったとする.

```
a=8    b=4
```

以下の文を実行した後の, a と b の値を示せ.

```
(a) a+=b;    (b) a-=b;    (c) a*=b;
(d) a/=b;    (e) b++;     (f) b--;
```

- [問 7] 以下の文は, プログラム中でどのような動作を行うか?

```
#define MAX_N 10000
```

2.2 制御文

例題と解答例のように, 問 1~問 4 の制御文を書け.

- [例題] a が -30 より大きい場合, 「end」と画面に書き出す.

[解等例]

```
if(-30 < a){
    printf("end\n");
}
```

- [問 1] a の値が 0 以下ならば, x の値を -1 倍する.
- [問 2] a の値が 10 以上かつ 100 以下ならば, b の値を 1 とする. さもなければ, b の値を -1 とする.
- [問 3] 以下のように, 1 つずつ値が増える整数と Hello world を 100 個表示する.

```
1    HelloWorld
2    HelloWorld
3    HelloWorld
    ⋮
100  HelloWorld
```

- [問 4] 以下のように, i の値が 1 ずつ増加したときの, i とその 2 乗と 3 乗の値を表示する. i の値は, 1~100 まで変化する.

```
1    1    1
2    4    8
3    9    27
    ⋮
100  10000 1000000
```

2.3 関数

- [問 1] リスト 1 のプログラムの実行した場合、ディスプレイに表示される文を示せ。
- [問 2] リスト 2 のプログラムの実行した場合、ディスプレイに表示される文を示せ。

リスト 1: 関数に関する問題 1

```
#include <stdio.h>
int hoge(int a, int b);
int fuga(int a, int b);

int main(void){
    int i, j, k, m;

    i=2;
    j=4;
    printf("i=%d\tj=%d\n", i, j);

    k = hoge(i+j, j+2);
    m = fuga(5, 4);

    printf("k=%d\tm=%d\n", k, m);

    return 0;
}

int hoge(int a, int b){
    int x;

    x = a*a + b*b;

    return x;
}

int fuga(int a, int b){
    int x;

    x = hoge(a+1, b+1);

    return x;
}
```

リスト 2: 関数に関する問題 2

```
#include <stdio.h>

void aswap(int a, int b);
void bswap(int *c, int *d);
void cswap(void);
void dswap(int dim []);

int e, f;

/* ----- main ----- */
int main(void){
    int a, b, c, d, g[2];

    a=1;
    b=2;
    c=3;
```

```
    d=4;
    e=5;
    f=6;
    g[0]=7;
    g[1]=8;

    aswap(a, b);
    printf("a = %d    b = %d\n", a, b);

    bswap(&c, &d);
    printf("c = %d    d = %d\n", c, d);

    cswap();
    printf("e = %d    f = %d\n", e, f);

    dswap(g);
    printf("g[0] = %d    g[1] = %d\n",
           g[0], g[1]);

    return 0;
}

/* ----- aswap ----- */
void aswap(int a, int b){
    int temp;

    temp=a;
    a=b;
    b=temp;
}

/* ----- bswap ----- */
void bswap(int *c, int *d){
    int temp;

    temp=*c;
    *c=*d;
    *d=temp;
}

/* ----- cswap ----- */
void cswap(void){
    int temp;

    temp=e;
    e=f;
    f=temp;
}

/* ----- dswap ----- */
void dswap(int dim []){
    int temp;

    temp=dim[0];
    dim[0]=dim[1];
    dim[1]=temp;
}
```

3 プログラム作成

3.1 基本

1~1000 までの和を計算し、その結果を表示するプログラムを作成せよ。

3.2 テイラー展開

三角関数の \sin をゼロの周りでテイラー展開すると、

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots$$

となる。この左辺と右辺の値を計算して比較する。以下の表のように、左端から角度 [deg]、 \sin の値、21 次までの値を表示するプログラムを作成せよ。

0	0.000000	0.000000
1	0.017452	0.017452
2	0.034899	0.034899
3	0.052336	0.052336
	⋮	
359	-0.017452	-0.017375
360	-0.000000	0.000083