

1 ファイルシステムに関する問題

1. 次の実行例の (a) から (h) に適切なパスを一つ答えなさい。(3 点× 8 問=24 点)

```
$ mkdir DirA DirB
$ echo aaa > DirA/a.txt
$ echo bbb > DirA/b.txt
$ echo ccc > DirA/c.txt
$ echo ddd > DirA/d.txt
$ ln -s (a) DirA/f.txt
$ cat DirA/f.txt
aaa
$ ln (b) DirA/e.txt
$ cat DirA/e.txt
aaa
$ cd DirA
$ ln (c) g.txt
$ cat g.txt
bbb
$ ln -s (d) h.txt
$ cat h.txt
bbb
$ rm a.txt b.txt
$ cat e.txt
(1)
$ cat f.txt
(2)
$ ln (e) ../DirB/a.txt
$ cat ../DirB/a.txt
ccc
$ ln -s (f) ../DirB/b.txt
$ cat ../DirB/b.txt
ddd
$ cd ../DirB
$ rm a.txt b.txt
$ ln (g) a.txt
$ cat a.txt
ccc
$ ln -s (h) b.txt
$ cat b.txt
ddd
$ rm ../DirA/c.txt ../DirA/d.txt
```

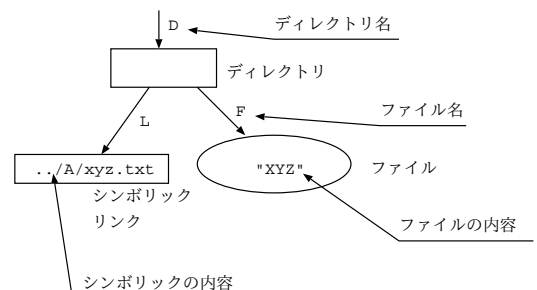
(a)	a.txt
(b)	DirA/a.txt (DirA/f.txt)
(c)	b.txt
(d)	b.txt (g.txt)
(e)	c.txt
(f)	../DirA/d.txt
(g)	../DirA/c.txt
(h)	../DirA/d.txt

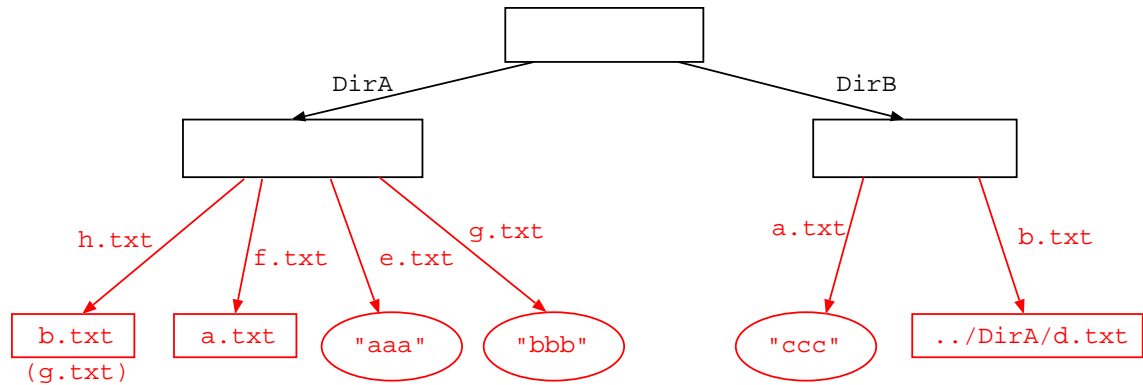
2. 前の実行例の (1),(2) 表示される内容、または、内容の説明を答えなさい。(3 点× 2 問=6 点)

(1)	aaa
(2)	f.txt ファイルが存在しないことを表すエラーメッセージが表示される

3. 実行後のファイル木の状態を記入例を参考に、次のページに描きなさい。(10 点)

記入例





2 シグナルに関する問題

次のプログラムを読んで答えなさい。

```
1  /*
2   * signal.c : シグナルの動作確認
3   */
4  #include <stdio.h>
5  #include <signal.h>
6
7  void h1(int sig) { printf("h1\n"); }
8  void h2(int sig) { printf("h2\n"); }
9  void h3(int sig) { printf("h3\n"); }
10 void h4(int sig) { printf("h4\n"); }
11
12 int main() {
13     signal(SIGHUP, h1);
14     signal(SIGTERM, h2);
15     signal(SIGINT, h3);
16     signal(SIGKILL, h4);
17
18     for(;;)
19         ;
20 }
```

1. 13行から16行の中で実行時にエラーになるシステムコールがあります。行番号とエラー原因を答えなさい。(8点)

行番号 : 16

理由 :

KILLシグナルはハンドリングの変更が許されないため。

2. 次のようにCtrl-Cを入力した時、画面に表示されるものを答えなさい。(3点)

```
$ signal
^C
```

h3

3. 次の実行結果の (a) から (c) に何を入力したか答えなさい。
(3 点×3 問=9 点)

```
$ signal
(a)
[4]+  Stopped                  signal
$ bg
[4]+  signal &
$ ps
  PID TTY          TIME CMD
 9526 ttys000    0:00.38 -bash
11801 ttys000    0:12.23 signal
$ (b)
h2
$ (c)
$ ps
  PID TTY          TIME CMD
 9526 ttys000    0:00.39 -bash
[4]+  Killed: 9                  signal
```

(a)	Ctrl-Z
(b)	kill 11801
(c)	kill -KILL 11801

3 高水準 I/O

5.1 に掲載したプログラムは、高水準 I/O の仕組みを説明するために作成した、簡易版の高水準 I/O ライブラリです。

myFILE.h と myFILE.c が簡易高水準 I/O ライブラリです。ここで定義された関数は標準 (stdio.h) の高水準 I/O を使用しないで、直接、システムコールを呼出します。

mycp3.c は、簡易高水準 I/O ライブラリの動作を確認するために、myFILE.h と myFILE.c の関数を使用して作った cp コマンドです。

次のようにコンパイルして使用します。

```
$ cc -o mycp3 mycp3.c myFILE.c
$ mycp3 file1 file2
```

1. プログラムリスト中の空欄 (a) から (e) に適切な記述を答えなさい。(3 点×5 問=15 点)

(a)	name, mod
(b)	fp->mode==O_RDONLY
(c)	myfflush(fp)
(d)	fp->bufCnt = n
(e)	MyFILE

2. mycp3.c の最後の myfclose() が省略できない理由を答えなさい。(5 点)

myfclose() を実行しないでプログラムを終了すると、myFILE 構造体の buf 配列内にある書き込み待ちのデータが、残る可能性がある。

プログラム終了前に myfclose() を実行して、バッファ内のデータを、確実に、ファイルに書き込む必要がある。

4 ファイル操作システムコール

プログラム (p1.c~p5.c) を読んで下の間に答えなさい。

```
// p1.c
#include <fcntl.h>
#include <unistd.h>

int main() {
    int fd = open("aaa",
                  O_WRONLY|O_CREAT,
                  0644);
    write(fd, "abc\n", 4);
    close(fd);
    return 0;
}
```

システムプログラミング2 H28 年度 前期中間試験 (2016.06.09 重村 哲至)

IE4 _____ 番 氏名 _____ 模範解答

(4/ 6)

```
// p2.c
#include <unistd.h>
int main() {
    link("aaa", "bbb");
    return 0;
}
```

```
// p4.c
#include <sys/stat.h>
int main() {
    chmod("bbb", 0750);
    return 0;
}
```

```
// p3.c
#include <unistd.h>
int main() {
    unlink("aaa");
    return 0;
}
```

```
// p5.c
#include <stdio.h>
int main() {
    rename("bbb", "ccc");
    return 0;
}
```

次の実行結果の (a) から (j) に表示されるものを答えなさい。(2 点× 10 問=20 点)

```
$ cc -o p1 p1.c; cc -o p2 p2.c; cc -o p3 p3.c; cc -o p4 p4.c; cc -o p5 p5.c;
$ mkdir A
$ cd A
$ ../p1
$ ls -l
-rw-r--r--  1 sigemura  staff  4 Jun  8 22:00 aaa
$ ../p2
$ ls -l
-rw-r--r-- (e)sigemura  staff  4 Jun  8 22:00 aaa
-rw-r--r-- (f)sigemura  staff  4 Jun  8 22:00 (a)
$ ../p3
$ ls -l
-rw-r--r-- (g)sigemura  staff  4 Jun  8 22:00 (b)
$ ../p4
$ ls -l
####(j)### (h)sigemura  staff  4 Jun  8 22:00 (c)
$ ../p5
$ ls -l
####(j)### (i)sigemura  staff  4 Jun  8 22:00 (d)
```

(a)	bbb	(b)	bbb	(c)	bbb	(d)	ccc
(e)	2	(f)	2	(g)	1	(h)	1
(i)	1	(j)	-rwxr-x---				

5 プログラムリスト

5.1 簡易高水準 I/O ライブラリ

```
/*
 * mycp3.c :
 * 簡易高水準 I/O ライブラリのテストドライバ
 * (cp コマンドを簡易高水準 I/O で作った例)
 */
#include <stdio.h> // fprintf のため
#include <stdlib.h> // exit のため
#include "myFile.h" // 簡易ライブラリのため

int main(int argc, char *argv[]) {
    if (argc!=3) {
        fprintf(stderr,
            "Usage: %s<srcFile> <dstFile>\n",
            argv[0]);
        exit(1);
    }

    #if 0
    *srcFp = myfopen(argv[1], "r");
    if (srcFp==NULL) {
        perror(argv[1]);
        exit(1);
    }

    #if 0
    *dstFp = myfopen(argv[2], "w");
    if (dstFp==NULL) {
        perror(argv[2]);
        exit(1);
    }

    int c;
    while ((c=mygetc(srcFp))!=MyEOF)
        myputc(c, dstFp);

    myfclose(srcFp);
    myfclose(dstFp); // 省略できない

    return 0;
}
```

```
/*
 * myFILE.h : 簡易高水準 I/O ライブラリの
 *            外部インタフェース
 */
#define BUFSIZ 1024 // バッファサイズ
#define MyEOF (-1) // 専用 EOF を定義

// 本ライブラリ専用の FILE 構造体
struct myFILE {
    int isBusy; // この構造体は使用中
    int fd; // 低水準 I/O の FD
    int mode; // オープンモード
    char buf[BUFSIZ]; // 読み書き用バッファ
    int bufPtr; // バッファの注目位置
    int bufCnt; // バッファ内バイト数
};
typedef struct myFILE MyFILE;

MyFILE *myfopen(char *name, char *modStr);
void myfclose(MyFILE *fp);
int myputc(char c, MyFILE *fp);
int mygetc(MyFILE *fp);
```

```
/*
 * myFILE.c : 簡易高水準 I/O ライブラリ本体
 */
#include <fcntl.h> // open
#include <unistd.h> // read, write, close
#include <string.h> // strcmp
#include <errno.h> // EINVAL, ENFILE
#include "myFILE.h" // 外部インタフェース

#define MAXFILES 10 // 同時最大オープン数

// 簡易高水準ファイル一覧表
MyFILE files[MAXFILES];

extern int errno; // エラー番号

// ファイルを開く : モードは "r", "w" の 2 種類
MyFILE *myfopen(char *name, char *modStr) {
    int mod = 0;
    if (strcmp(modStr, "r")==0)
        mod = O_RDONLY;
    else if (strcmp(modStr, "w")==0)
        mod = O_WRONLY|O_CREAT|O_TRUNC;
    else {
        errno = EINVAL; // 「無効なモード」
        return NULL; // という意味のエラー
    }
    for (int i=0; i<MAXFILES; i++) {
        if (files[i].isBusy==0) {
            int fd = open( # (a) # , 0644);
            if (fd<0) return NULL;
            files[i].isBusy = 1;
            files[i].fd = fd;
            files[i].mode = mod;
            files[i].bufPtr = 0;
            files[i].bufCnt = 0;
            return &files[i];
        }
    }
    errno = ENFILE; // 「ファイル数多すぎ」
```

```
    return NULL; // という意味のエラー
}

// バッファのデータをファイルに出力する
static void myfflush(MyFILE *fp) {
    if ( # (b) # ) return;
    write(fp->fd, fp->buf, fp->bufCnt);
    fp->bufPtr = 0;
    fp->bufCnt = 0;
}

// ファイルを閉じる
void myfclose(MyFILE *fp) {
    myfflush(fp);
    fp->isBusy = 0;
}

// ファイルに 1 文字出力する
int myputc(char c, MyFILE *fp) {
    fp->buf[fp->bufPtr++] = c;
    fp->bufCnt++;
    if (fp->bufCnt >= BUFSIZ) # (c) # ;
    return c;
}

// バッファにファイルからデータを入力する
static void myfill(MyFILE *fp) {
    int n = read(fp->fd, fp->buf, BUFSIZ);
    fp->bufPtr = 0;
    # (d) # ;
}

// ファイルから 1 文字入力する
int mygetc(MyFILE *fp) {
    if (fp->bufCnt == fp->bufPtr) myfill(fp);
    if (fp->bufCnt <= 0) return MyEOF;
    int c = fp->buf[fp->bufPtr++];
    return c & 0xff;
}
```