

システムプログラミングⅡ 令和6年度 前期中間試験

(2024.06.04 重村 哲至)

IE4

番 氏名

模範解答

1 語句に関する問題

空欄に最適な言葉を語群から記号で答えなさい。

(1点×20問=20点)

ファイルの読み書きをする (1) には, read, write がある。これらを直接に使用する入出力は (2) 入出力と呼ばれる。fput などの (3) 入出力関数は, データを (4) にためて1回の write でまとめて出力する (5) を行う。

標準入出力ストリームは, プログラム起動前に (6) することで, キーボードやディスプレイだけではなくファイルの読み書きもできる。このように, キーボードやディスプレイのような入出力装置もファイルも, 同じプログラムで扱うことができるのは (7) I/O のお陰である。

ファイルはストレージに記録された (8) のデータ記憶である。ファイルはディレクトリにより (9) 的に管理される。ファイルとディレクトリによる (9) 構造は (10) と呼ばれる木構造で表現される。 (11) は木構造の根に, 他の (12) は木構造の節に, (13) は木構造の葉に, (14) は木構造の有効枝に対応する。

(10) 中のファイルやディレクトリは (15) により特定することができる。相対 (15) は (16) ディレクトリを, 絶対 (15) は (17) ディレクトリを起点にする。

(18) はプロセスに非同期的にイベントの発生を知らせる仕組みである。プロセスが (18) を受信したときの取り扱いには, 終了, 無視, (19) がある。 (19) されるのは, 予め (20) が登録されたシグナルである。

語群:

- (あ) カレント, (い) シグナルハンドラ関数,
(う) シグナル, (え) システムコール,
(お) ディレクトリ, (か) パス, (き) バッファリング,
(く) バッファ, (け) ファイル, (こ) ファイル木,
(さ) ユニファイド, (し) リダイレクト, (す) リンク,
(せ) ルート, (そ) ルートディレクトリ, (た) 階層,
(ち) 高水準, (つ) 低水準, (て) 不揮発性,
(と) 補足 (キャッチ)

(1)	(え)	(2)	(つ)	(3)	(ち)	(4)	(く)
(5)	(き)	(6)	(し)	(7)	(さ)	(8)	(て)
(9)	(た)	(10)	(こ)	(11)	(そ)	(12)	(お)
(13)	(け)	(14)	(す)	(15)	(か)	(16)	(あ)
(17)	(せ)	(18)	(う)	(19)	(と)	(20)	(い)

2 低水準入出力

付録の mydd プログラムは dd プログラムの簡易版です。使用方法は次の書式のとおりで。

書式 mydd infile outfile bsize count

引数 infile: 入力ファイルのパス
outfile: 出力ファイルのパス
bsize: ブロックサイズ
count: ブロック数

解説 mydd プログラムは, infile のデータを outfile にコピーします。コピーするデータは, infile の先頭から bsize バイトのデータブロック count 個です。infile の長さが bsize × count バイトより短い場合は, infile の最後まで outfile にコピーして終了します。

mydd プログラムは実行を終了する際に, コピーしたデータブロックの個数と, コピーしたデータの合計バイト数を標準エラー出力に表示します。mydd プログラムは, outfile が既に存在する場合は一旦ファイルを空にした上でデータを書き込みます。outfile が存在しない場合は保護モードが rw-r--r-- の新しいファイルを作ります。mydd プログラムは, ファイルのオープンでエラーが発生した場合, オープンに失敗したファイルのパスを含むエラーメッセージを表示します。mydd プログラムは簡易版なので, read/write でエラーが発生しても正しいエラー処理ができないかもしれません。

1. mydd.c 中の空欄 (#(?)#) に適切なプログラムの記述を答えなさい。(3点×10問=30点)

システムプログラミングⅡ 令和6年度 前期中間試験

(2024.06.04 重村 哲至) IE4 番 氏名 模範解答

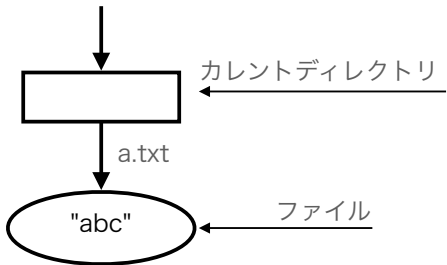
(1)	exit(1)
(2)	!= 5
(3)	O_RDONLY
(4)	argv[1]
(5)	O_WRONLY O_TRUNC O_CREAT
(6)	0644
(7)	argv[2]
(8)	<= 0
(9)	n
(10)	stderr

2. 実行例1の空欄(#(?)#)に何が表示されたか答えなさい。なお、#(2)#はファイルサイズである。
(4点×3問=12点)

(1)	-rw-r--r--
(2)	320
(3)	01234567

3 ファイル操作

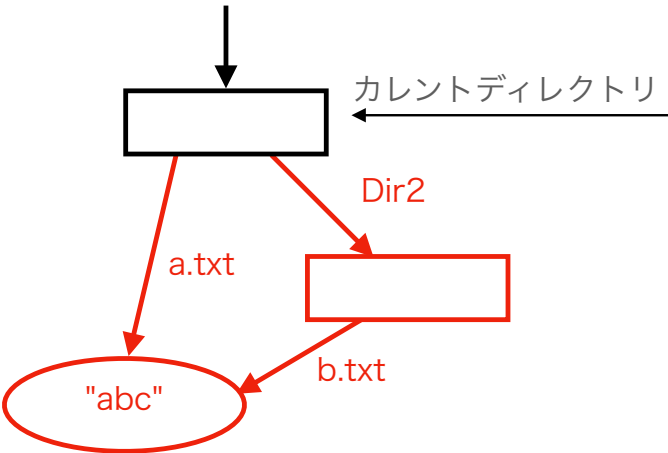
次の図のように、カレントディレクトリ以下に a.txt ファイルだけがあります。次の問に答えなさい。



1. 付録の p1.c を実行したときと同じ結果になる操作をコマンドラインで実行します。コマンドラインに入力する6つのコマンドを実行順に書きなさい。(4点×6問=24点)

(1)	mkdir Dir1
(2)	mkdir Dir2
(3)	ln a.txt Dir1/a.txt
(4)	ln Dir1/a.txt Dir2/b.txt
(5)	rm Dir1/a.txt
(6)	rmdir Dir1

2. p1.c を実行したあとのカレントディレクトリ以下の様子を図で表しなさい。(5点)



4 シグナル

付録の mysleep.c は、授業で出題した課題6の解答です。myslpmain.c は課題6の同名のプログラムを書き換えたものです。問に答えなさい。

- mysleep.c 中(1)の行が無かった場合, mysleep(0); を実行するとどうなるか答えなさい。(3点)
mysleep() が永遠に終わらなくなる。
- myslpmain の実行結果(表示されるもの)を答えなさい。(3点)
3秒経過
6秒経過
- mysleep.c 中の(2)行の org を SIG_DFL にした場合の, myslpmain の実行結果(表示されるもの)を答えなさい。(3点)
3秒経過

付録

```
// mydd.c
// #include は省略
void error_exit(char *myname) {
    fprintf(stderr,
        "使用方法: %s infile outfile bsize count\n",
        myname);
    #(1)#;
}
int main(int argc, char *argv[]) {
    if (argc #(2)#) {
        error_exit(argv[0]);
    }
    // atoi は文字列"123"を数値123に変換する
    int bsize = atoi(argv[3]);
    int count = atoi(argv[4]);
    if (bsize <= 0 || count <= 0) {
        error_exit(argv[0]);
    }
    // 入力ファイルを開く
    int infd = open(argv[1], #(3)#);
    if (infd < 0) {
        perror(#(4)#);
        return 1;
    }
    // 出力ファイルを開く
    int outfd = open(argv[2], #(5)#, #(6)#);
    if (outfd < 0) {
        perror(#(7)#);
        return 1;
    }
    char buf[bsize];
    int bytes = 0;
    int blocks = 0;
    for (blocks=0; blocks<count; blocks++) {
        int n = read(infd, buf, bsize);
        if (n #(8)#) break;          // EOF or Error
        write(outfd, buf, #(9)#);
        bytes = bytes + n;
    }
    close(infd);
    close(outfd);
    fprintf(#(10)#, "%d blocks, %d bytes\n",
        blocks, bytes);
    return 0;
}
```

実行例 1

```
% ls -l aaa
ls: aaa: No such file or directory
% ./mydd /dev/urandom aaa 16 20
20 blocks, 320 bytes
% ls -l aaa
```

```

#(1)#  1 sige  staff  #(2)# Jun  2 23:01 aaa
% echo 0123456789 > 0.txt
% ls -l 0.txt
-rw-r--r--  1 sige  staff  11 Jun  2 23:03 0.txt
% cat 0.txt
0123456789
% ./mydd 0.txt 1.txt 1 8
8 blocks, 8 bytes
% ls -l 1.txt
####  1 sige  staff  8 Jun  2 23:03 1.txt
% cat 1.txt
#(3)#
```

```
// p1.c
// #include は省略
int main(int argc, char *argv[]) {
    mkdir("Dir1", 0755);
    mkdir("Dir2", 0755);
    link("a.txt", "Dir1/a.txt");
    link("Dir1/a.txt", "Dir2/b.txt");
    unlink("Dir1/a.txt");
    rmdir("Dir1");
    return 0;
}
```

```
// mysleep.c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include "mysleep.h"
static void alHandler(int n) { }
unsigned int mysleep(unsigned int seconds) {
    if (seconds==0) return 0;          // (1)
    sig_t org = signal(SIGALRM, alHandler);
    alarm(seconds);
    pause();
    unsigned int r = alarm(0);
    signal(SIGALRM, org);              // (2)
    return r;
}
```

```
// myslpmain.c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include "mysleep.h"
static void alarmHandler(int n) { }
int main() {
    signal(SIGALRM, alarmHandler);
    mysleep(3);
    printf("3秒経過\n");
    alarm(3);
    pause();
    printf("6秒経過\n");
    return 0;
}
```