

システムプログラミングⅡ 令和2年度 前期中間試験

(2022.06.06 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)  
実行中のユーザプログラムのことを(1)と呼ぶ。(1)は(2)を発行しOS(カーネル)に処理を依頼する。  
高水準I/Oは、(3)にデータを溜め(4)システムコールの回数を少なくする。fopen関数は(5)システムコールを発行しファイルを開いた上で、FILE構造体を割り付ける。  
ファイルは(6)に格納された(7)のデータ記憶である。  
ファイルを格納するファイル木は、(8)ディレクトリを根とする有向の木構造である。ファイル木の中でユーザの現在位置は(9)ディレクトリ、ログインした直後の(9)ディレクトリは(10)ディレクトリ、あるディレクトリから見て根に近い側のディレクトリは(11)ディレクトリと呼ばれる。ファイル木でファイルの位置はパスによって表現できる。(8)ディレクトリを起点とするパスは(12)パス、(9)ディレクトリを起点とするパスは(13)パスと呼ばれる。  
シグナルは(1)に(14)的にイベントの発生を知らせせるOSの仕組みである。(1)はシグナルの扱いであるシグナル(15)を予め決めることができる。シグナル(15)は(16)システムコールを用いてシグナルの種類毎に決める。シグナルを(17)するためにはハンドリング関数を(16)に渡す。シグナルを(18)するためにはハンドリング関数のかわりに定数SIG\_IGNを渡す。(1)が他の(1)にシグナルを送信するには(19)システムコールを用いる。システムコールで相手(1)の指定には(20)を用いる。

語群：(あ)kill, (い)open, (う)read/write, (え)signal, (お)カレント, (か)システムコール, (き)バッファ, (く)ハンドリング, (け)プロセス, (こ)プロセス番号, (さ)ホーム, (し)ルート, (す)親, (せ)絶対, (そ)相対, (た)二次記憶装置(ストレージ), (ち)同期, (つ)非同期, (て)不揮発性, (と)補足, (な)無視

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

2 実行結果

1. 次の実行例で最初の5行は正常に実行できるものとします。6行目以降でエラーになる行の右の四角に「×」、そうではない行に「○」を記入しなさい。(2点×16問=32点)

|                             |   |
|-----------------------------|---|
| \$ mkdir A                  |   |
| \$ cd A                     |   |
| \$ mkdir X                  |   |
| \$ mkdir Y                  |   |
| \$ echo abc > X/a.txt       |   |
| \$ ln X/a.txt Y/1.txt       | ○ |
| \$ ln ../X/a.txt Y/2.txt    | × |
| \$ ln -s X/a.txt Y/3.txt    | ○ |
| \$ ln -s ../X/a.txt Y/4.txt | ○ |
| \$ cat Y/1.txt              | ○ |
| \$ cat Y/2.txt              | × |
| \$ cat Y/3.txt              | × |
| \$ cat Y/4.txt              | ○ |
| \$ mv Y/1.txt ../1.txt      | ○ |
| \$ mv Y/2.txt ../2.txt      | × |
| \$ mv Y/3.txt ../3.txt      | ○ |
| \$ mv Y/4.txt ../4.txt      | ○ |
| \$ cat 1.txt                | ○ |
| \$ cat 2.txt                | × |
| \$ cat 3.txt                | ○ |
| \$ cat 4.txt                | × |

2. 付録の実行例1の空欄( # (?) # )の表示を答えなさい。(3点×6問=18点)

|     |                 |
|-----|-----------------|
| (1) | 4               |
| (2) | -rwxr--r--      |
| (3) | -rw-rw-r--      |
| (4) | /Users/sige/A/B |
| (5) | /Users/sige     |
| (6) | /Users/sige     |

## システムプログラミングⅡ 令和2年度 前期中間試験

(2022.06.06 重村 哲至)

IE4

\_\_\_\_ 番 氏名

模範解答

3. 付録のプログラム p1.c を読み, 実行例 2 の空欄 (#(?)#) の表示を答えなさい. (3 点 × 3 問 = 9 点)

|     |   |
|-----|---|
| (1) | <code>-rw-r--r--</code>                 |
| (2) | <code>11</code>                         |
| (3) | <code>abc</code><br><code>abcabc</code> |

4. 付録のプログラム p2.c, p3.c を読み, 実行例 3, 4 の空欄 (#(?)#) の表示を答えなさい.  
(3 点 × 2 問 = 6 点)

|     |                    |
|-----|--------------------|
| (1) | <code>b.txt</code> |
| (2) | <code>b.txt</code> |

5. 次は付録のプログラム p4.c の実行例です. ターミナル 1 で実行した p4 をターミナル 2 だけ操作して終了させました. 空欄に入力したコマンドを答えなさい. (5 点)

### ターミナル 1

```
$ ./p4
Ctrl-Cが押されるまで待ちます
Ctrl-Cが押されました
$
```

### ターミナル 2

```
$ ps
  PID TTY          TIME CMD
 46752 ttys001    0:00.00 ./p4
 91249 ttys001    0:00.65 -zsh
 46754 ttys002    0:00.05 -zsh
$ ###(1)###
$
```

|     |                              |
|-----|------------------------------|
| (1) | <code>kill -INT 46752</code> |
|-----|------------------------------|

## 3 プログラムの完成

次のプログラムの 2 つの printf の間に 3 行追加し, 最初の printf の実行後 5 秒間スリープし, 第 2 の printf を実行して終了するように完成しなさい. (10 点)

```
// p5.c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
void h(int n){}
int main() {
    printf("5 秒待ちます\n");

    signal(SIGALRM, h);

    alarm(5);

    pause();

    printf("5 秒経過しました\n");
    return 0;
}
```

## システムプログラミングⅡ 令和2年度 前期中間試験

(2022.06.06 重村 哲至)

IE4

\_\_\_\_ 番 氏名

模範解答

### 付録

#### 実行例1

```
$ echo bbb > b.txt
$ ls -l b.txt
-rw-r--r-- 1 sige staff #(1)# Jun 4 21:00 b.txt
$ chmod u+x b.txt          # uは所有者の意味
$ ls -l b.txt
###(2)### 1 sige staff #(1)# Jun 4 21:00 b.txt
$ chmod 664 b.txt
$ ls -l b.txt
###(3)### 1 sige staff #(1)# Jun 4 21:00 b.txt
$ pwd
/Users/sige/A
$ mkdir B
$ cd B
$ pwd
###(4)###
$ cd ../../
$ pwd
###(5)###
$ cd .
$ pwd
###(6)###
$
```

#### 実行例2

```
$ ls -l a.txt
ls: a.txt: No such file or directory
$ ./p1 a.txt
$ ls -l a.txt
###(1)### 1 sige staff #(2)# Jun 4 22:25 a.txt
$ cat a.txt
###(3)###          # 複数行の可能性あり
$
```

#### 実行例3

```
$ ls *.txt
a.txt  b.txt
$ ./p2 a.txt b.txt
$ ls *.txt
###(1)###
$
```

#### 実行例4

```
$ ls *.txt
```

```
a.txt  b.txt
$ ./p3 a.txt b.txt
$ ls *.txt
###(2)###
$
```

```
// p1.c
#include <fcntl.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    int fd=open(argv[1],
                 O_WRONLY|O_CREAT|O_TRUNC,
                 0644);
    write(fd, "abc\n", 4);
    write(fd, "abc\n", 3);
    write(fd, "abc\n", 4);
    close(fd);
    return 0;
}
```

```
// p2.c
#include <unistd.h>
int main(int argc, char *argv[]) {
    unlink(argv[2]);
    link(argv[1], argv[2]);
    unlink(argv[1]);
    return 0;
}
```

```
// p3.c
#include <stdio.h>
int main(int argc, char *argv[]) {
    rename(argv[1], argv[2]);
    return 0;
}
```

```
// p4.c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
void h(int n){}
int main() {
    signal(SIGINT, h);
    printf("Ctrl-Cが押されるまで待ちます\n");
    pause();
    printf("Ctrl-Cが押されました\n");
    return 0;
}
```