

I 以下の a ~ f の文章の空欄を埋めるのに最も適切な語句を下の候補の中から選んで解答用紙に書きなさい。同じ候補を複数回使っても構いません。(42 点)

- a. CPU やメモリなどハードウェアや、ファイル名、アドレスなど、計算機システム中に限られた数や量しか用意されていないものを、一般に (1) と呼び、カーネルの主要な役割はこれらを管理することである。CPU の (2) モードで動作しているユーザープロセスは、通常、これらに直接アクセスできないため、(3) を行って、カーネルに仕事を依頼する必要がある。
- b. OS の主要な機能として、プロセス管理、メモリ管理、(4) 管理、デバイス管理がある。
- c. 仮想記憶システムを備えた OS の場合、ユーザープロセスの機械語プログラムが使用するアドレスを (5) アドレスと呼び、CPU のアドレス変換機構により変換されて (6) 装置に伝えられるアドレスを (7) アドレスと呼ぶ。これにより、異なる (8) の同一の仮想アドレスを異なる物理アドレスに対応させることができる。
- d. バイトアドレッシングが採用されている場合、4 B 長のデータ 0x11223344 をリトルエンディアン方式で 0x1234 番地から主記憶装置に格納すると、0x1235 番地に格納される 1 B のデータは (9) となる。
- e. (10) マルチタスク方式の OS では、ユーザープロセスが自発的に CPU の割り当を放棄することにより、プロセスのスケジューリングを行うが、(11) マルチタスク方式では、タイマーからの (12) 信号を利用してプロセスのスケジューリングを行う。
- f. ユーザープロセスの実行中に割り込みが発生すると、カーネルプログラムへの (13) スイッチが起こり、CPU は (14) モードに移行して、その割り込み対する処理が行われる。

空欄の候補

0x00, 0x11, 0x12, 0x22, 0x23, 0x33, 0x34, 0x44, 0x45, ALU, TLB, アイドル待ち, イベント待ち, カーネル, コンテキスト, システムコール, シェル, シンボリックリンク, スタック, スタックポインタ, サーバ, テキスト, ディレクトリ, ハードリンク, ヒープ, ブートストラップ, ファイル, プリエンプティブ, プログラムカウンタ, プロセス, ライブラリ, リターン, 演算, 階層的, 仮想, 協調的, 資源, 実行可能, 実行中, 主記憶, 数理, 静的, 動的, 特権, 入出力, 非特権, 物理, 補助記憶, 命令レジスタ, 例外, 割り込み

II 1 ページの大きさが 1 KiB であるようなページング方式の仮想記憶システムを持ったオペレーティングシステムが稼働しており、ユーザープロセスが 1 つだけ動作している。このプロセスの仮想アドレス空間は 0x0000 番地 ~ 0x3fff 番地で、16 個の仮想ページで構成されている。一方、このプロセスが使用可能な物理ページは、ページ番号 0x1 から 0x7 までの 7 ページだけである。ある時点で、このプロセスの仮想ページ番号と物理ページ番号が右の表のように対応していた。ただし、表の空欄は対応する物理ページが存在しないことを表している。物理ページが不足する場合、最後にアクセスされた時刻が最も遠い過去であるようなページをページアウトするものとして問いに答えなさい。(25 点)

- (1) この時点で、仮想アドレス 0x35a4 番地はどの物理アドレスに対応しているか。
- (2) この後、このプロセスが仮想ページ番号 0xa にアクセスするとページフォルトが発生するが、その処理が終って、この仮想ページに対応することになる物理ページ番号を答えなさい。
- (3) (2) に続いて、このプロセスが、次の順に仮想ページ番号にアクセスしたとする。最後にページアウトした仮想ページ番号を答えなさい。

0x3, 0xd, 0x8, 0xc, 0x4, 0xf, 0x3, 0x7, 0x1, 0xb, 0x6, 0x9, 0x5, 0x2, 0x3, 0xd, 0xe, 0x5 0xc, 0xd, 0xb, 0x5, 0x2

- (4) (3) のとき、最後にページインした仮想ページ番号を答えなさい。

仮想ページ番号	物理ページ番号
0x0	
0x1	
0x2	0x7
0x3	0x1
0x4	
0x5	
0x6	0x5
0x7	
0x8	0x2
0x9	
0xa	
0xb	
0xc	
0xd	0x4
0xe	0x3
0xf	

(裏面に問題 III)

Ⅲ Unix 系 OS 上の端末エミュレータにおいて、下のようにコマンドを実行した。a ~ f の文章の空欄を埋めるのに最も適切な語句を下の候補の中から選んで解答用紙に書きなさい。ただし、foo.c の内容は右のような C プログラムであり、シンボリックリンクは使用されていないものとします。同じ候補を複数回使っても構いません。(33 点)

- foo を起動して、stddev 関数の機械語プログラムの先頭番地へ分岐が起こる際には、(1) アドレスが、このプロセスのスタックにプッシュされる。
- foo の実行時に、配列 data は、このプロセスの (2) 領域に割り当てられる。
- このプログラムが直接呼び出しているライブラリ関数は scanf と printf、(3) の 3 つである。これらの関数の機械語プログラムは、foo というファイルには含まれておらず、実行時に (4) リンクされる。
- foo を実行すると、printf 関数は (5) というシステムコールを使って、プロセスの (6) ヘータを書き出すが、このデータは (7) を介して端末エミュレータに送られる。
- ./foo <sample の実行時に、sample というファイルを open したプログラムは (8) である。math グループに属する t190000 というユーザーは、この sample というファイルに対して (9) を行うことができない。
- (10) ディレクトリを起点としたパス名の書式を絶対パス名と呼ぶ。このソースファイル foo.c を絶対パス名で最も簡潔に示すと (11) となる。

```
foo.c
#include <stdio.h>
#include <math.h>

double data[100];
int n = 0;

double stddev(int n) {
    int i;
    double s = 0.0, d = 0.0, m;

    for (i = 0; i < n; i++)
        s += data[i];
    m = s / n;
    for (i = 0; i < n; i++)
        d += (data[i]-m) * (data[i]-m);
    return sqrt(d / n);
}

int main() {
    double d;

    while (n < 100) {
        if (scanf("%lf", &d) != 1)
            break;
        data[n++] = d;
    }
    if (n > 0)
        printf("%f\n", stddev(n));
    return 0;
}
```

端末エミュレータでのコマンド実行の様子

```
$ pwd
/home/y230000
$ cd test
$ cc -o foo ../../math/foo.c -lm
$ ls -l
合計 20
-rw--wxr-x 1 y230000 math 152 7月 31 15:02 sample
-rwxrwxr-x 1 y230000 math 16920 8月 2 11:01 foo
$ ./foo <sample
11.161347
$
```

空欄の候補

foo.c、math/foo.c、test/foo.c、/math/foo.c、/home/foo.c、/home/math/foo.c、/home/y230000/foo.c、/home/y230000/math/foo.c、/home/y230000/test/foo.c、ALU、BSS、CPU、TLB、close、main、mmap、open、read、foo、sqrt、stddev、write、カレント、コンテキスト、シェル、シンボリック、スケジューリング、スタック、テキスト、データ、パイプ、ハッシュ、ハード、ヒープ、ファイル、プッシュ、プログラムヘッダ、プロセス、ポップ、リターン、ルート、ロック、親、階層的、書き込み、仮想、疑似端末、子、実行、数値、静的、動的、標準エラー出力、標準出力、標準入力、物理、読み出し、例外、割り込み

- I (1) 資源 (2) 非特権
- (3) システムコール (4) ファイル
- (5) 仮想 (6) 主記憶
- (7) 物理 (8) プロセス
- (9) 0x33 (10) 協調的
- (11) プリエンプティブ (12) 割り込み
- (13) コンテキスト (14) 特権

II (1) $0x35a4 = 0011\ 0101\ 1010\ 0100_{(2)} = 0xd \times 1024 + 0x1a4$ であるから、このアドレスは仮想ページ番号 $0xd$ に属し、与えられた対応表によると、物理ページ番号 $0x4$ に対応する。よって、求める物理アドレスは $0x4 \times 1024 + 0x1a4 = 0001\ 0001\ 1010\ 0100_{(2)} = 0x11a4$ となる。

(2) このプロセスが使用可能な物理ページ番号は $0x1 \sim 0x7$ で、 $0x6$ だけが空いているので、このページが使用される。つまり、求める物理ページ番号は $0x6$ である。

(3) と (4) の解答欄は裏面

情報処理 (計算機) システム II ・ 定期試験 ・ 解答用紙

(II の解答欄の続き)

(3) アクセスした仮想ページ番号を直近から遠い過去に向かって並べると、

0x2、0x5、0xb、0xd、0xc、0xe、0x3、0x9、0x6、...

の順となるが、使用できる物理ページは7ページだけなので、8番目となっている物理ページ番号0x9が最後にページアウトしたはずである。

(4) (3) で、仮想ページ0x9がページアウトして以降の仮想ページへのアクセス順は、

0x5、0x2、0x3、0xd、0xe、0x5 0xc、0xd、0xb、0x5、0x2

であり、この中で最も遅く登場する0xbが求める仮想ページ番号となる。

- III (1) リターン (2) BSS
- (3) sqrt (4) 動的
- (5) write (6) 標準出力
- (7) 疑似端末 (8) シェル
- (9) 読み出し (10) ルート
- (11) /home/math/foo.c