情報処理システム II ・中野・定期試験問題

2025 年 8 月 1 日

I 以下の a ～ e 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。(45 点)

a. ユーザープロセスの機械語プログラムは、CPU の (1) モードで実行されるため、カーネルが管理している様々な (2) に直接アクセスすることはできない。このため、(3) 割り込みを発生させる特別な機械語命令を実行することで、カーネルに対して必要な仕事を依頼することになる。この割り込みで、CPU は (4) モードとなり、カーネルがあらかじめ設定しておいた機械語プログラムへの (5) スイッチが起り、そこでカーネルはユーザープロセスから依頼された仕事を実行する。この仕事を直ちに完了することができな場合、ユーザープロセスは、その完了までは (6) 待ち状態となる。

b. 仮想記憶システムが採用されている場合、同じ番地を使用する複数のユーザープロセスが同時に存在していても問題が生じないのは、カーネルが、CPU の (7) 機構による (8) アドレスから (9) アドレスへの変換のされかたをユーザープロセスごとに切り替えるためである。

c. 入出力装置などからの割り込みが発生すると、カーネルによって、その処理が行われる。例えば、(10) マルチタスク方式の OS では、(11) からの割り込みを利用して、ユーザープロセスから CPU の割り当てを強制的に取り上げることができるため、ユーザープロセスが (12) 的に CPU の割り当てを放棄しなくても、プロセスのスケジューリングを行うことができる。

d. Unix 系 OS の `pipe` というシステムコールは、信頼性のある (13) 通信を行う際の端点となる 2 つの (14) を作成する。

e. 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 をビッグ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、0x1234 番地に格納される 1 B のデータは (15) となる。

空欄の候補

`0x00`、`0x11`、`0x12`、`0x22`、`0x23`、`0x33`、`0x34`、`0x44`、`0x45`、`ALU`、`TLB`、アイドル、アドレス、アドレス変換、イベント、カーネル、コンテクスト、コンテクスト変換、システムコール、シェル、スタック、スタックポインタ、ソケット、ソフトウェア、サーバ、タイマー、テキスト、データグラム、ディレクトリ、バイトストリーム、パケット、ハードウェア、ヒープ、ファイル、ファイル記述子、ブートストラップ、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、仮想、強制、資源、実行可能、実行中、自発、主記憶、数理、静的、線形変換、動的、特権、入出力、配列、非特権、物理、補助記憶、命令レジスタ、例外、割り込み

Ⅱ 1 MiB の主記憶装置 (物理メモリ) を持つ計算機システムで、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、右の C プログラムをコンパイルして得られる機械語プログラムを実行する。このプロセスは、配列 data のための領域として (1 MiB の内の) 1000 KiB の物理メモリを常に占有しており、配列 data へのアクセス以外でページフォルトが起こることはない。ページフォルトが起こると、(必要なら) 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物理ページを (補助記憶装置に) ページアウトし、(必要なら) アクセスしたい仮想ページの内容を (補助記憶装置から) その物理ページにページインする。int 型のデータの大きさは 4 B、配列要素 data[0] の仮想アドレスが 0x12400 であるとして、以下の問いに答えなさい。ただし、バイトアドレッシング方式で、キャッシュメモリの影響は考えないものとする。(25 点)

```
 1 #define N (500*1024)
 2
 3 int data[N];
 4 int sum;
 5
 6 int main()
 7 {
 8     while (1) {
 9         int i;
10
11         for (i = 0; i < N; i += 64) {
12             sum += data[      ];
13             sum -= data[i];
14         }
15     }
16     return 0;
17 }
```

(1) 配列 data 全体が占める仮想ページ数を答えなさい。

(2) data[64] のアドレス (4 B の領域の先頭アドレス) の仮想ページ番号とオフセットをそれぞれ 16 進数で答えなさい。

(3) 12 行目の空欄を i + 32 で埋めたとする。このプログラムの実行中に、13 行目の data[i] へのアクセスでページフォルトが起こる確率 (割合) を求めなさい。

(4) 12 行目の空欄を N/2 で埋めたとする。このプログラムの実行中に、13 行目の data[i] へのアクセスでページフォルトが起こる確率 (割合) を求めなさい。

(5) 12 行目の空欄を (i + N/2) % N で埋めたとする。このプログラムの実行中に、13 行目の data[i] へのアクセスでページフォルトが起こる確率 (割合) を求めなさい。

(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次ページ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する。同じ候補を複数回使っても構いません。(30 点)

foo.c

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>

#define N (1000)

char buf[N];

int main() {
    int fd, len;

    fd = open("in.data", O_RDONLY);

    if (fd < 0) {
        write(2, "open failed!\n", 13);
        return 1;
    }

    len = read(fd, buf, N);

    if (len == 0)
        return 0;

    if (len < 0) {
        read(2, "read failed!\n", 13);
        return 1;
    }

    if (write(1, buf, len) != len) {
        write(2, "write failed!\n", 14);
        return 1;
    }

    return 0;
}
```

a. foo を起動すると、配列 buf は、このプロセスの仮想アドレス空間の [(1)] 領域に、変数 len は [(2)] 領域に割り当てられる。

b. また、関数 main の機械語プログラムは [(3)] 領域に置かれるが、この領域の仮想ページは、通常、[(4)] 不可に設定されており、同時に実行されている同じプログラム間で同じ [(5)] ページを共有できるようになっている。

c.「./foo >out.data」の実行時には、シェルは [(6)] というシステムコールで自分自身のプロセスを複製し、新しくできたプロセスが標準 [(7)] を一旦 [(8)] した上で、out.data というファイルを open し、その後 [(9)] というシステムコールにより、foo が起動する。

d.「./foo >out.data」の実行が正常に終了すると、このプロセスの終了コードは [(10)] となり、新しいファイル out.data の大きさは [(11)] バイトとなるはずである。

e. この実行例では問題となっていないが、「./foo >out.data」の実行時に、もし、[(12)] ディレクトリの in.data というファイルに対する読み出し許可がない場合、このプロセスの標準 [(13)] には、「[(14)] failed!」という文字列が書き出されるはずである。

f. この実行例で使用されているソースファイル foo.c を相対パス名で最も簡潔に示すと [(15)] となる。

端末エミュレータでのコマンド実行の様子

```
$ pwd
/home/y240000
$ cc -o foo /home/nakano/src/foo.c
$ ls -l
合計 20
-rw---x--x 1 y240000 math     14  7月 31 15:02 in.data
-rwxrwxr-x 1 y240000 math  16808  8月  1 11:01 foo
$ ./foo >out.data
$
```

空欄の候補

0、1、2、3、12、13、14、20、1000、foo.c、./foo.c、../foo.c、../src/foo.c、../nakano/src/foo.c、src/foo.c、nakano/src/foo.c、/home/y240000/foo.c、/home/nakano/src/foo.c、ALU、BSS、CPU、TLB、close、execve、_exit、fork、main、mmap、open、pipe、read、wait、write、エラー出力、カレント、コンテクスト、シェル、シンボリック、スタック、テキスト、データ、パイプ、ハード、ヒープ、ファイル、プッシュ、プロセス、リターン、ルート、ロック、親、階層的、書き込み、仮想、実行、数理、静的、動的、出力、入力、物理、読み出し、例外、割り込み

(定期試験問題終り)

情報処理システム II ・定期試験・解答用紙

I (1) 非特権　(2) 資源

(3) ソフトウェア　(4) 特権

(5) コンテクスト　(6) イベント

(7) アドレス変換　(8) 仮想

(9) 物理　(10) プリエンプティブ

(11) タイマー　(12) 自発

(13) バイトストリーム　(14) ファイル記述子

(15) `0x11`

II (1) 4 B の大きさの要素が 500×1024 個並んだものであるから、$4\ \mathrm{B} \times 500 \times 1024 = 2000$ KiB、つまり、2000 ページとなる。

(2) 1 つの要素が 4 番地分に相当するため、$\mathrm{0x12400} + (4 \times 64) = \mathrm{0x12400} + 256 = \mathrm{0x12400} + \mathrm{0x100} = \mathrm{0x12500} = 0001\ 0010\ 0101\ 0000\ 0000_{(2)}$
1 ページが 1 KiB で、この下位 10 b がオフセットとなるから、ページ番号は $0100\ 1001_{(2)} = \mathrm{0x49}$ で、オフセットは $01\ 0000\ 0000_{(2)} = \mathrm{0x100}$ となる。

(3) `i` が 64 増えると、`data[i]` のアドレスは 256 番地増えるが、1 ページが 1024 B であるため、12 行目の `data[i + 32]` と 13 行目 `data[i]` は、同じページへのアクセスとなる。つまり、13 行目のアクセスの直前に同じページにアクセスしており、このページはすでにページインしているはずなので、求める確率は 0 となる。

(4) と (5) の解答欄は裏面

学籍番号　氏名　(裏面に続く)

情報処理システム II ・定期試験・解答用紙

(II の解答欄の続き)

(4) 13 行目の data[i] のアドレスは 256 番地ずつ増えて、4 回に 1 回、ページ番号が 1 つ大きいページへ進む。大きさ 2000 ページの配列 data に対して割り当てられる物理ページは 1000 ページしかないので、このページが 12 行目でアクセスする data[N/2] と同じページであれば、ページフォルトは起きないが、そうでなければページフォルトが発生する。よって、求める確率は

$$\frac{1999}{4 \times 2000} = \frac{1999}{8000}$$

(5) 12 行目でアクセスするページと 13 行目でアクセスするページは (2000 を法として) 1000 ページ分のずれを保ちつつ、それぞれ、4 回に 1 回は新しいページに進む。一方、大きさ 2000 ページの配列 data に対して割り当てられる物理ページは 1000 ページしかないので、どちらのアクセスも 4 回に 1 回はページアウトしているページにアクセスすることになる。よって、求める確率は

$$\frac{2000}{4 \times 2000} = \frac{1}{4}$$

III (1) BSS　(2) スタック

(3) テキスト　(4) 書き込み

(5) 物理　(6) fork

(7) 出力　(8) close

(9) execve　(10) 0

(11) 14　(12) カレント

(13) エラー出力　(14) open

(15) ../nakano/src/foo.c

(解答用紙終り)