情報処理 (計算機) システム II ・中野・金 2 ・定期試験問題

2023 年 7 月 28 日

I ある Unix 系 OS の計算機環境にログインし、端末エミュレータを開き、下図左のような入力をした。ただし、foo.c の内容は下図右の通りである。下の候補の中から最も適当と思われる語句を選んで空欄を埋めなさい。(51 点)

```
$ cc -o foo foo.c
$ ./foo
x = 12
12*12 = 144
x = 1
1*1 = 1
x = -1
$
```

foo.c

```
 1 #include <stdio.h>
 2
 3 int x;
 4
 5 int main() {
 6     while (1) {
 7         printf("x = ");
 8         scanf("%d", &x);
 9         if (x < 0)
10             break;
11         printf("%d*%d = %d\n", x, x, x*x);
12     }
13     return 0;
14 }
```

a. シェルは、キーボードから入力された「cc -o foo foo.c」という文字列を、 (1) というシステムコールを使って読み取ると、 (2) という (あるいはそれに類する) システムコールを使って自分のプロセスを複製する。このとき生成された子プロセスは、さらに (3) というシステムコールを使って cc の機械語プログラムの実行を開始する。その後、親のプロセス (シェルのプロセス) は、子プロセス (cc のプロセス) が終了するまでは (4) 状態となる。

b. foo.c をコンパイルすると、7 行目や 8 行目、11 行目の関数呼び出し式に対しては、分岐命令の一種である (5) 命令が生成される。この機械語命令は、それに続く命令が置かれたアドレス (リターンアドレス) を、 (6) にプッシュして、指定されたアドレスに分岐する。プッシュされたリターンアドレスは、呼び出された関数の機械語プログラムの中で (7) 命令が実行された際にポップされ、分岐先のアドレスとして使用される。

c. foo.c の 3 行目で宣言されている変数 x の値を記憶するためのメモリ領域は、通常、このプロセスの (8) 領域にとられる。8 行目の式 &x の値は、このメモリ領域の先頭の (9) アドレスであり、CPU のアドレス変換機構によって (10) アドレスに変換されて使われる。

d. foo.c の 8 行目で 呼び出されている関数 scanf は、このプロセスの (11) (ファイル記述子 0) から読み込んだ文字列を int 型の値に変換し、このアドレスに書き込む仕事を行う。

e. foo.c の 7 行目や 11 行目で呼び出されている関数 printf は、 (12) というシステムコールを使って、「x = 」や「12*12 = 144」などの文字列を、このプロセスの (13) (ファイル記述子 1) に出力する。printf の機械語プログラムは、このプロセスの (14) 用領域に置かれているはずである。

f. この foo のプロセスを含めて、ユーザープロセスは CPU の (15) モードで実行されるため、計算機システムの資源に直接アクセスすることはできない。このため、カーネルに対して (16) を行って、これを代行してもらう必要がある。

g. Unix 系 OS では (17) マルチタスク方式が採用されているため、たとえ、foo.c に不具合があって、その実行が無限ループに陥ったとしても、他のユーザープロセスに CPU 資源を割り当てることができる。

空欄の候補

BSS、close、connect、dup、execve、fork、mmap、open、read、TLB、unlink、write、イベント待ち、カーネル、キャッシュ待ち、コンテクスト、システムコール、スタック、ストア、テキスト、ディレクトリ、パイプライン、ヒープ、ファイル、プリエンプティブ、プロセス、ロード、演算、仮想、疑似端末、協調的、実行待ち、実行中、数理、特権、非特権、標準入力、標準出力、標準エラー出力、物理、復帰、呼び出し、割り込み

(次ページに問題 II と III)

Ⅱ 2 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 2 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 d のための領域として (2 MiB の内の) 2000 KiB の物理メモリを常に占有しており、配列 d 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。double 型のデータの大きさは 8 B、配列要素 d[0] の仮想アドレスが 0x15800 であるとして、以下の問いに答えなさい。ただし、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 S (  A  )
 2 #define N (S*1024)
 3
 4 double d[N];
 5
 6 int main() {
 7     int i;
 8     double sum = 0.0;
 9
10     while (1) {
11         for (i = 0; i < N; i += [ B ])
12             sum += d[i];
13     }
14     return sum;
15 }
```

(1) d[10] のアドレス (8 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。

(2) A の空欄を 1000 で、B の空欄を 8 で埋める。このプログラムの実行中に、12 行目の d[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。

(3) A の空欄を正の整数 s で、B の空欄を 8 で埋める。このプログラムの実行開始後、程なくすると、12 行目の d[i] へのアクセスでページフォルトが起こらないようになった。このような s の最大値を答えなさい。

(4) A の空欄を 1000 で、B の空欄を正の整数 p で埋める。このプログラムの実行開始後、程なくすると、12 行目の d[i] へのアクセスでページフォルトが起こらないようになった。このような p の最小値を答えなさい。

Ⅲ math グループのみに属している a89023 というユーザが、ある Unix 系 OS の計算機環境にログインし、端末エミュレータを開き、次のようにコマンドを入力した。空欄を埋めるのに適切な語句を解答用紙に書きなさい。ただし、シンボリックリンクは使用されていないものとする。(24 点)

```
$ pwd
/mnt/test
$ cd ../progs
$ ./bar >result <../data
処理が終了しました
$ ls -l
合計 44
-r-xrwxrw- 1   a89023 math   237  7月 25 10:15 bar.c
---xrwxr-- 1  y220000 math 16808  7月 25 10:18 bar
-rwxr-xrwx 1  y220000 elec   225  7月 28 11:02 result
$
```

a. bar の実行時に、その標準入力となっているファイルを絶対パス名で最も簡潔に示すと [(1)] となる。

b. カレントディレクトリが /usr/etc である場合、bar を相対パス名で最も簡潔に示すと [(2)] となる。

c. bar.c、bar、result の 3 つのファイルの内、このユーザが書き込み権を持っているのは [(3)] と [(4)] である。

d. bar の実行時に、本来なら端末エミュレータへ送られる bar の出力は、あるファイルに書き込まれる。そのファイルを絶対パス名で最も簡潔に示すと [(5)] となる。この時、書き込まれたデータの大きさは [(6)] バイトである。

e. bar を実行した際に、端末エミュレータに「処理が終了しました」と表示されたのは、bar がこの文字列を、標準 [(7)] と呼ばれる [(8)] 番のファイル記述子に書き込んだためである。

(定期試験問題終り)

情報処理 (計算機) システム II ・定期試験・解答用紙

I

(1) read　(2) fork

(3) execve　(4) イベント待ち

(5) 呼び出し　(6) スタック

(7) 復帰　(8) BSS

(9) 仮想　(10) 物理

(11) 標準入力　(12) write

(13) 標準出力　(14) mmap

(15) 非特権　(16) システムコール

(17) プリエンプティブ

II

(1) d[10] の仮想アドレスは 0x15800 + (8 × 10) = 0x15850 となり、これを二進表記すると

0001 0101 1000 0101 0000

となる。一方、ページサイズが $2048 = 2^{11}$ であるから、0001 0101 1 が仮想ページ番号で、000 0101 0000 がオフセットとなる。それぞれ、16 進数で表記すると、仮想ページ番号が 0x2b、オフセットが 0x50 となる。

(2) 〜 (4) の解答欄は裏面

学籍番号　氏名　**(裏面に続く)**

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)

(2) 配列要素 `d[i]` のアドレスは $8\times8=64$ B 分ずつ増えていき、配列のすべてのページへのアクセスが発生する。配列全体の大きさは $8\times1000\times1024$ B で、1 ページが 2048 B であるから、4000 ページに相当するが、使用可能な物理メモリは 2000×1024 B、つまり 1000 ページなので、配列全体の大きさに満たない。ページの境界を跨いで、新しいページにアクセスする場合、そのページは最も遠い過去にアクセスしたページであるから、ページアウトされており、ページフォルトが発生する。このプログラムでは、同じページに $2048\div64=32$ 回アクセスするが、残りの 31 回では、すでにページインされているので、ページフォルトは発生しないはずである。よって、求める確率は $\frac{1}{32}$ と考えられる。

(3) 配列要素 `d[i]` のアドレスは $8\times8=64$ B 分ずつ増えていくので、配列 `d` が割り当てられるすべての仮想ページに順にアクセスすることになる。このため、ページフォルトが起らないようになるための必要十分条件は、この配列が使用する仮想ページ数が、占有できている物理ページ数以下になることである。よって、この配列の要素数の最大値は $\frac{2000\times1024}{8}=250\times1024$ であり、求める s の最大値は $\frac{250\times1024}{1024}=250$。

(4) 配列全体の大きさは 8000 KiB となり、2000 KiB の物理メモリに収まらない。このため、ページフォルトが起らなくなるための必要十分条件は、この配列全体の仮想ページの $\frac{1}{4}$ 以下にしかアクセスしないということになる。つまり、1 回ごとに `d[i]` のアドレスが 4 ページ分大きくなればよい。よって、求める p の最小値は $\frac{4\times2048}{8}=1024$。

Ⅲ (1) `/mnt/data` (2) `../../mnt/progs/bar`

(3) `bar` (4) `result`

(5) `/mnt/progs/result` (6) `225`

(7) エラー出力 (8) `2`

(解答用紙終り)