

オペレーティングシステム I H30 年度 後期末試験

(2019.02.13 重村 哲至)

IE4 _____ 番 氏名

模範解答

1 語句に関する問題

次の文章の空欄に最適な言葉を語群から記号で答えなさい。ただし、語群で A,B,C,D は空欄の番号が指定されているので注意すること。(2点×20問=40点)

プロセス間通信

プロセス間で共有メモリを用いた情報の交換が可能である。共有メモリはそれぞれのプロセスの (1) メモリ空間に同じ (2) メモリを関係付けることで実現できる。関係付けるためには (3) を使用するが、関係付が完了したら (3) を使用することなく情報の交換ができる。しかし、共有メモリにプロセス間の (4) は無いので、別途セマフォなどを用いて競合を避ける必要がある。

メッセージ通信は (3) を用いてプロセス間でデータをコピーする方式である。通信相手の指定方法には、直接指定方式と間接指定方式がある。直接指定方式では通信先として (5) を指定する、間接指定方式では通信先として (6) を指定する、多対多の通信に向いているのは (7) 指定方式の方である。

(8) を用いることでメッセージ通信を使用するプロセスの待ち時間を短くできる。例えば送信プロセスは、(8) に空きがあればデータを書き込んですぐに次の処理に移ることができる。一方で (8) に空きが無いときは、空きができるまでシステムコールがブロックする (9) 方式と、即座にシステムコールがエラーで完了する (10) 方式を選択できる。

メッセージ通信で用いるメッセージには、メッセージの種類を表すための (11) を付加できる場合がある。受信プロセスは (11) を指定してメッセージを選択的に受信できる。

モニタ

モニタは同期のための機能が予め組み込まれた (12) である。モニタは内部に (13) を持つことがで

きる。(13) はモニタの外部から直接アクセスすることができない。モニタは外部から呼び出されるプログラム「手続き」を持つことができる。手続きは (14) の働きにより排他的に実行される。(15) には wait と signal の操作ができる。wait 操作をしたプロセスは (15) の待ち行列に入る。

複数資源の同時使用

複数の資源を同時に必要とするような場合は (16) の発生に注意が必要である。資源を順位付けし、順位に従った順番で確保することで (16) の発生を回避することが可能である。このようにすることで資源待ちプロセスの関係が整理され、(16) 発生の原因の一つである (17) が発生しないようにできる。

しかし、「食事する哲学者の問題」のように順位付けが難しい場合もある。(16) のもう一つの発生原因は、一部の資源を確保した状態で他の資源の待ち状態なる (18) である。(17) と (18) のどちらかが発生しないようにすることは、(16) の発生が防止するために (19) である。(18) をしないために、複数の資源を同時に確保する特別な命令を準備する。セマフォの場合なら (20) 命令がそれに当たる。

語群:(あ) P_and, (い) wait_or, (う) ガード, (え) システムコール, (お) タグ, (か) デッドロック, (き) バッファ, (く) プロセス, (け) リンク, (こ) 確保待ち, (さ) 仮想, (し) 循環待ち, (す) 資源(データ, 変数), (せ) 条件変数, (た) 抽象データ型, (ち) 同期, (つ) 同期機構, (て) 非同期, (と) 物理, (A) 直接(7用), (B) 間接(7用), (C) 十分(19用), (D) 不十分(19用)

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

オペレーティングシステム I H30 年度 後期末試験

(2019.02.13 重村 哲至)

IE4

____ 番 氏名

模範解答

2 モニタ

次は教科書に掲載されていた仮想言語のモニタで記述したキューです。よくプログラムを読んで間に答えなさい。

```
1 monitor BoundedBuffer {
2     // 資源(リングバッファ)
3     int N;
4     int[] buf;
5     int first, last, cnt;
6     // 条件変数
7     Condition empty;
8     Condition full;
9     // 初期化
10    BoundedBuffer(int n) {
11        N = n;
12        buf = new int[N];
13        first = last = cnt = 0;
14    }
15    // 手続き
16    public void append(int x) { // (a)
17        if (cnt==N) full.wait(); // (b)
18        buf[last] = x; // (c)
19        last = (last + 1) % N; // (d)
20        cnt++; // (e)
21        empty.signal(); // (f)
22    }
23    public int remove() { // (g)
24        if (cnt==0) empty.wait(); // (h)
25        int x = buf[first]; // (i)
26        first = (first + 1) % N; // (j)
27        cnt--; // (k)
28        full.signal(); // (l)
29        return x; // (m)
30    }
31 }
```

(1) 行と実行順序を記号で答えなさい。

キューが空の時、次の順でキューが操作されたとします。リスト中の (a)~(m) の行が実行される順序を記号で答えなさい。

1. スレッド 1 が remove() を呼び出す。
2. スレッド 2 が append() を呼び出す。

a. remove() 呼び出し時 (5 点)

(g) → (h)

b. append() 呼び出し時 (5 点)

(a) → (b) → (c) → (d) → (e) → (f)
→ (i) → (j) → (k) → (l) → (m)

(2) 行と実行順序を記号で答えなさい。

N が 1 でキューが空の時、次の順で操作されたとします。リスト中の (a)~(m) の行が実行される順序を記号で答えなさい。

1. スレッド 1 が append() を呼び出す。
2. スレッド 2 が append() を呼び出す。
3. スレッド 3 が remove() を呼び出す。

a. 1 回目の append() 呼び出し時 (5 点)

(a) → (b) → (c) → (d) → (e) → (f)

b. 2 回目の append() 呼び出し時 (5 点)

(a) → (b)

c. remove() 呼び出し時 (5 点)

(g) → (h) → (i) → (j) → (k) → (l) → (c) → (d) → (e) → (f) → (m)

オペレーティングシステム I H30 年度 後期末試験

(2019.02.13 重村 哲至)

IE4

____ 番 氏名

模範解答

3 モニタの応用

次はセマフォを仮想言語のモニタで記述したものです。プログラムと解説を読んで問に答えなさい。

```
1 monitor Semaphore {
2     // 資源(セマフォの状態)
3     int val;           // セマフォの値
4     int cnt;           // 待ちプロセス数
5
6     // 条件変数
7     Condition c;
8
9     // 初期化
10    Semaphore(int n) {
11        val = n;
12        cnt = 0;
13    }
14
15    // 手続き
16    public void P() {    // P 操作
17        if (val>0) {
18            val--;
19        } else {
20            cnt++;
21            #####
22            cnt--;
23        }
24    }
25
26    public void V() {    // V 操作
27        if (cnt<=0) {
28            #####
29        } else {
30            #####
31        }
32    }
33 }
```

解説

モニタの条件変数には、スレッドを条件変数の待ち行列に入れる `wait()` と、待ち行列のスレッドを実行可能にする `signal()` の2つの操作だけができるものとします。 `signal()` は実行可能になったスレッドを即座に実行させます。

記述されたセマフォには `P()` 操作と `V()` 操作をすることができます。 `P()` 操作はセマフォの値 (`val`) が1以上なら、値を減らして処理を終わります。 `P()` 操作はセマフォの値が0以下なら、スレッドを条件変数の待ち行列に入れます。 `cnt` は条件変数の待ち行列に入っているスレッドの数を表しています。

`V()` 操作は条件変数の待ち行列が空ならセマフォの値を1増やします。 `V()` 操作は条件変数の待ち行列にスレッドが入っていれば、その中から一つを実行可能にします。

(1) プログラム中の空欄 (#####) に適切な記述を答えなさい。

- a. 21 行 (5 点)
`c.wait();`
- b. 28 行 (5 点)
`val++;`
- c. 30 行 (5 点)
`c.signal();`

(2) プログラム中の行番号で答えなさい。

初期化により、セマフォの状態を `val=0`, `cnt=0` にして、以下を順に実行したとします。

- a. まず、スレッド1が `P()` 操作を行った。どの行でブロックするか? (5 点)
21 行
- b. a. に続いてスレッド2が `V()` 操作を行った。28行と30行のどちらが実行されるか? (5 点)
30 行

オペレーティングシステム I H30 年度 後期末試験

(2019.02.13 重村 哲至)

IE4

____ 番 氏名

模範解答

4 共有メモリ

プロセス間の共有メモリとスレッド間の共有変数の違いについて述べなさい。(図などを用いて説明してもよい。)
(10 点)

スレッド間の共有変数は、同じプロセス内のスレッドが共有する変数であり、同じ仮想メモリ空間に配置されるメモリ領域のことである。

プロセス間の共有メモリは、異なるプロセスの間で共有されるので、複数の仮想メモリ空間から共有されるメモリ領域のことである。

単一の仮想メモリ空間に配置されるか、複数の仮想メモリ空間に横断的に配置されるかが違う。

