OS1 H29 年度 後期末試験

(2018.02.08 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．なお，(24)，(25) は，(な)，(に) のどちらかで答えなさい．(1 点× 25 問= 25 点)

複数のスレッドが並行して実行される時，ある **(1)** に対して操作を行うプログラムの部分を **(2)** と呼ぶ．**(2)** は複数のスレッドによって並行して実行されてはならない．もしも，**(2)** を並行して実行すると **(1)** に対する **(3)** が発生する．**(3)** が発生すると正しい結果が得られない可能性があるので，スレッド間で **(1)** に対する **(4)** を行う必要がある．**(2)** の入口と出口で **(4)** を行うために必要になる処理を行うプログラムの部分を **(5)** (入口側)，**(6)** (出口側) と呼ぶ．シングルプロセッサシステムでは **(5)** で **(7)** を実行し，**(6)** で **(8)** を実行すれば目的が達成できる．

セマフォは **(9)** とスレッドの **(10)** を持つデータ型である．セマフォに対して **(9)** の値をデクリメントする **(11)** とインクリメントする **(12)** を行うことができる．**(9)** の値がゼロのセマフォに **(11)** を行うとスレッドは **(13)** し **(10)** に追加される．

(14) は複数のプロセスで同じ物理メモリを参照する方式である．この方式はプロセス間の **(4)** 機構や **(15)** 機構は提供しない．**(16)** はシステムコールを用いてプロセス間で情報をコピーする方式である．通信相手の指定方式には相手プロセスのプロセス番号等を用いる **(17)** 指定方式とリンクを作成し同じリンクを指定するプロセス同士で通信する **(18)** 指定方式がある．

モニタは資源管理用の機能と制約を持つ特別な **(19)** である．モニタは外部から呼出して実行できる **(20)** を持つことができる．**(20)** は **(21)** の働きにより同時には一つしか実行されない．モニタの内部の資源は外部からは **(20)** を通してしかアクセスできない．モニタは状態変数を持つことができる．条件変数には，待ち行列にプロセスを追加する **(22)** 操作と，待ち行列のプロセスを起床させる **(23)** 操作をすることができる．**(22)** 操作を行ったプロセスは **(21)** を **(24)** ブロックする．**(23)** 操作を行った時に起床するプロセスがある場合は，**(21)** を **(25)** 起床したプロセスの実行に切り換わる．

語群： (あ) wait, (い) signal, (う) P 操作, (え) V 操作, (お) エグジットセクション, (か) エントリーセクション, (き) カウンタ, (く) ガード, (け) クリティカルセクション, (こ) ブロック, (さ) メッセージ通信, (し) 間接, (す) 競合, (せ) 共有メモリ, (そ) 資源, (た) 相互排除, (ち) 抽象データ型, (つ) 直接, (て) 手続き, (と) 同期, (な) 外さないで (24, 25), (に) 外してから (24, 25), (ぬ) 待ち行列, (ね) 割込み禁止命令, (の) 割込み許可命令

(1)	(そ)	(2)	(け)	(3)	(す)	(4)	(た)	(5)	(か)	(6)	(お)
(7)	(ね)	(8)	(の)	(9)	(き)	(10)	(ぬ)	(11)	(う)	(12)	(え)
(13)	(こ)	(14)	(せ)	(15)	(と)	(16)	(さ)	(17)	(つ)	(18)	(し)
(19)	(ち)	(20)	(て)	(21)	(く)	(22)	(あ)	(23)	(い)	(24)	(に)
(25)	(な)										

2 専用命令による相互排除

次のような二つの機械語命令がある。(以下で z はゼロフラグの意味とする。) マルチプロセッサシステムでも使用できる相互排除の仕組みをリスト 1 に示す.

TS (Test and Set) 命令

```
TS R,M
```

1. バスをロックする
2. $R \leftarrow [M]$
3. if (R==0) $z \leftarrow 1$; else $z \leftarrow 0$;
4. $[M] \leftarrow 1$
5. バスのロックを解除する

SW (Swap) 命令

```
SW R,M
```

1. バスをロックする
2. $T \leftarrow [M]$
3. $[M] \leftarrow R$
4. $R \leftarrow T$
5. バスのロックを解除する

リスト 1: TS 命令を用いた相互排除

```
// エントリーセクション
L1      DI
        TS      G0, FLG
        JZ      L2
        EI
        JMP     L1

// クリティカルセクション
L2      ...

// エグジットセクション
        LD      G0, #0
        ST      G0, FLG
        EI

// 非クリティカルセクション
        ...

// 初期値ゼロフラグ
FLG     DC      0
```

1. TeC 風の命令が使用できるものとし，リスト 1 のエントリーセクションを SW 命令を用いて書き換えなさい。(ヒント：CMP 命令はレジスタとメモリの値を比較する) (5 点)

```
L1      DI
        LD      G0, #1
        SW      G0, FLG
        CMP     G0, #0
        JZ      L2
        EI
        JMP     L1
```

2. TeC 風の命令が使用できるものとし，リスト 1 のエントリーセクションを CAS 命令を用いて書き換えなさい。但し、CAS 命令は次のような命令とする。(5 点)

CAS（Compare And Swap）命令

`CAS R0,R1,M`

(a) バスをロックする

(b) $T \leftarrow [M]$

(c) `if` $(T == R0)$ { $[M] \leftarrow R1$; $z \leftarrow 1$; } `else` { $R0 \leftarrow T$; $z \leftarrow 0$; }

(d) バスのロックを解除する

```
L1      DI
        LD      G0, #0
        LD      G1, #1
        CAS     G0, G1, FLG
        JZ      L2
        EI
        JMP     L1
```

3. なぜクリティカルセクションを割込みを禁止で実行する必要があるか説明しなさい。(5 点)

- クリティカルセクションで何が発生しないようにしているか？
  プリエンプションの発生を防いでいる。
- それが発生すると次に何が起こる可能性があるか？
  プリエンプションが発生し優先度の高いプロセスに実行が切り替った場合、優先度の高いプロセスがエントリーセクションでビジーウェイティングを始めることがある。
- 最終的にどのような状態になる可能性があるのか？
  その場合、プリエンプションした優先度の低いプロセスに実行が移らず、デッドロックになる可能性がある。

3 セマフォによる相互排除

複数のプロセスが変数 val の値をインクリメント・デクリメントする場合に，正しく処理ができるようにセマフォを用いた相互排除を行った例を示します。

```
int       val;
Semaphore S;
void inc() {
  (a)
  val++;
  (b)
}
void dec() {
  (c)
  val--;
  (d)
}
```

1. セマフォ（S）の初期値はいくつにするべきか？(3 点)

   1

2. プログラム中、(a)、(b)、(c)、(d) に適切なセマフォ操作を答えなさい。ただし、セマフォの操作は P(S)、V(S) のように書くことにします。(3 点× 4 問=12 点)

(a)	P(S);	(b)	V(S);
(c)	P(S);	(d)	V(S);

4 セマフォの使用順序

三つの資源 A、B、C から二つを使用する三つのプロセス P_1、P_2、P_3 がある。三つの資源それぞれに資源利用の相互排除を行うためのセマフォSa、Sb、Sc がある。

P_1
```
1:P(Sa);
2:P(Sb);
3:資源 A、B を使用する
4:V(Sb);
5:V(Sa);
```

P_2
```
1:P(Sb);
2:P(Sc);
3:資源 B、C を使用する
4:V(Sc);
5:V(Sb);
```

P_3
```
1:P(Sc);
2:P(Sa);
3:資源 C、A を使用する
4:V(Sa);
5:V(Sc9);
```

1. 三つのプロセスが同時に 1:行を実行した場合、何が発生するか？(4 点)

   デッドロック

2. P_1、P_2、P_3 で、1 と同様なことが発生する実行順を示しなさい（ない場合も含む）。(4 点)

   他にはない

3. 1 が発生しないような P_1、P_2、P_3 の改良例を一つ答えなさい。(4 点)

   P_3 の 1 行と 2 行の内容を入れ替え、セマフォの確保順を Sa→Sb→Sc に統一する。

5 セマフォによるリーダライタ問題

次は C 言語風の言語で記述した，リーダライタ問題のセマフォによる解を示しています。

```
Data      records;                  // 共有するデータ
Semaphore rwSem;                    // リーダとライタの排他用セマフォ
void writerThread() {               // ライタスレッド(複数のスレッドで並列実行する)
  for ( ; ; ) {
    Data d = produce();             // 新しいデータを作る
    _(a)_;
    writeRecores( d );              // データを書換える
    _(b)_;
  }
}
int       cnt = 0;                  // リーダ間の共有変数(読出し中のリーダ数)
Semaphore cntSem;                   // cnt の排他制御用セマフォ
void readerThread() {               // リータスレッド(複数のスレッドで並列実行する)
  for ( ; ; ) {                     // リーダスレッドは以下を繰り返す
    _(c)_;
    if ( cnt == 0 ) _(d)_;
    cnt = cnt + 1;                  // cnt をインクリメント
    _(e)_;
    Data d = readRecords();         // データを読みだす
    _(f)_;
    cnt = cnt - 1;                  // cnt をデクリメント
    if ( cnt == 0 ) _(g)_;
    _(h)_;
    consume( d );                   // データを使用する
  }
}
```

1. プログラム中の_(a)_から_(h)_に適切なセマフォ操作を答えなさい。(2 点× 8 問=16 点)

(a)	P(rwSem);	(b)	V(rwSem);
(c)	P(cntSem);	(d)	P(rwSem);
(e)	V(cntSem);	(f)	P(cntSem);
(g)	V(rwSem);	(h)	V(cntSem);

2. 二つのセマフォの初期値はそれぞれいくつにすべきか答えなさい。(2 点× 2 問=4 点)

rwSem	1	cntSem	1

6 モニタによるセマフォ実装

次は授業で紹介した仮想言語のモニタを用いてセマフォを実現した例です。条件変数には c.wait() と c.signal() の操作ができるものします。

```
1  monitor Semaphore {
2    int val;           // セマフォの値
3    Condition c;       // 条件変数
4    int cnt;           // 待ちプロセス数
5    // 初期化プログラム
6    Semaphore(int n) {
7      val = n;
8      cnt = 0;
9    }
10   // P 操作
11   public void P() { // ガードに守られる
12     if (_(a)_) {
13       val--;
14     } else {
15       _(b)_;
16       c.wait();
17       cnt--;
18     }
19   }
20   // V 操作
21   public void V() { // ガードに守られる
22     if (_(c)_) {
23       val++;
24     } else {
25       c.signal();
26     }
27   }
28 }
```

1. プログラム中_(a)_から_(c)_に適切なプログラムを答えなさい。(3 点× 3 問=9 点)

(a)	val>0
(b)	cnt=cnt+1
(c)	cnt<=0

2. 一つ目のプロセスが P() を実行し 16 行の wait() でブロックしたとき、二つ目のプロセスが V() を実行し 25 行の signal() を実行した。17、18、26 行の実行順を答えなさい。(4 点)

17行 → 18行 → 26行の順に実行される。