

オペレーティングシステム I 令和5年度 後期末試験

(2024.02.14 重村 哲至)

IE4

番 氏名

模範解答

注意：以下で「プロセス」と「スレッド」は同じ意味で用いられていることがある。

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。(1点×30問=30点)

ヒント：語群の言葉を全て1回使用します。

複数のスレッドが資源を共有して処理を進める場合、資源の利用について適切な(1)を行わないと(2)が発生することがある。(2)が発生するプログラムの部分は(3)と呼ばれる。(3)の入口である(4)で(3)に入る権利を取得し、出口である(5)で権利を返却する。

シングルプロセッサシステムでは、スレッドが(6)しないように(3)を(7)禁止状態で実行することで(1)を行うことができる。マルチプロセッサシステムでは、(8)のような特別な命令を用いる。この方法では(4)で(9)を行うので長時間の待ちになる場合には適さない。

セマフォは(10)とスレッドの(11)を含むデータ型である。セマフォに対して(10)を減らす(12)と、(10)を増やす(13)をすることができる。(10)がゼロのセマフォに対して(12)を行ったスレッドは(14)になる。

プロセス間で情報を共有する代表的な機構として共有メモリとメッセージ通信がある。共有メモリ使用を開始する前に(15)を用いて準備作業をする必要があるが、準備が完了すれば(15)を使用しないでデータの交換ができる。

メッセージ通信はデータ交換のたびに(15)を使用する方式である。通信相手の指定方式には、相手のプロセス番号などを用いる(16)方式と、(17)の番号などを用いる(18)方式がある。また、データが届くまで(15)がブロックする方式は(19)方式、ブロックしない方式は(20)方式と呼ばれる。

モニタの(21)は、複数のスレッドが同時にモニタ内部の手続きを実行できないようにする機構である。モニタ内部の資源や条件変数は、モニタ外部のプログラムから直接操作することはできない。モニタ内部の手続きは条件変数に対して(22)、(23)の2つの操作ができる。(23)を行ったスレッドはモニタの(21)を(24)、条件

変数の待ち行列に入る。(22)を行うと条件変数の待ち行列から一つのスレッドが選ばれ(25)になる。

(26)は、複数のスレッドがあったとき、互いを待ち合い処理が先に進まない状態のことである。(26)の原因は、資源を確保したスレッドが追加で別の資源を必要とするときに発生する(27)、スレッドが互いに相手が確保している資源が必要になったときに発生する(28)等である。(27)を発生させないためには、必要な資源を全て一括して確保する方法がある。セマフォの操作を改良した(29)等を用いることで一括確保ができる。(28)を発生させないためには、資源の(30)に制約を設ける方法がある。

語群：

- (あ) P_AND, (い) P 操作, (う) signal, (え) TS 命令,
(お) V 操作, (か) wait, (き) エグジットセクション,
(く) エントリーセクション, (け) ガード,
(こ) カウンタ, (さ) クリティカルセクション,
(し) システムコール, (す) デッドロック,
(せ) ビジーウェイティング, (そ) プリエンプション,
(た) リンク, (ち) 確保順序, (つ) 確保待ち,
(て) 間接指定, (と) 競合, (な) 実行可能,
(に) 循環待ち, (ぬ) 相互排除, (ね) 直接指定,
(の) 同期, (は) 外して(解除して), (ひ) 非同期,
(ふ) 待ち行列, (へ) 待ち状態, (ほ) 割込み,

(1)	(ぬ)	(2)	(と)	(3)	(さ)
(4)	(く)	(5)	(き)	(6)	(そ)
(7)	(ほ)	(8)	(え)	(9)	(せ)
(10)	(こ)	(11)	(ふ)	(12)	(い)
(13)	(お)	(14)	(へ)	(15)	(し)
(16)	(ね)	(17)	(た)	(18)	(て)
(19)	(の)	(20)	(ひ)	(21)	(け)
(22)	(う)	(23)	(か)	(24)	(は)
(25)	(な)	(26)	(す)	(27)	(つ)
(28)	(に)	(29)	(あ)	(30)	(ち)

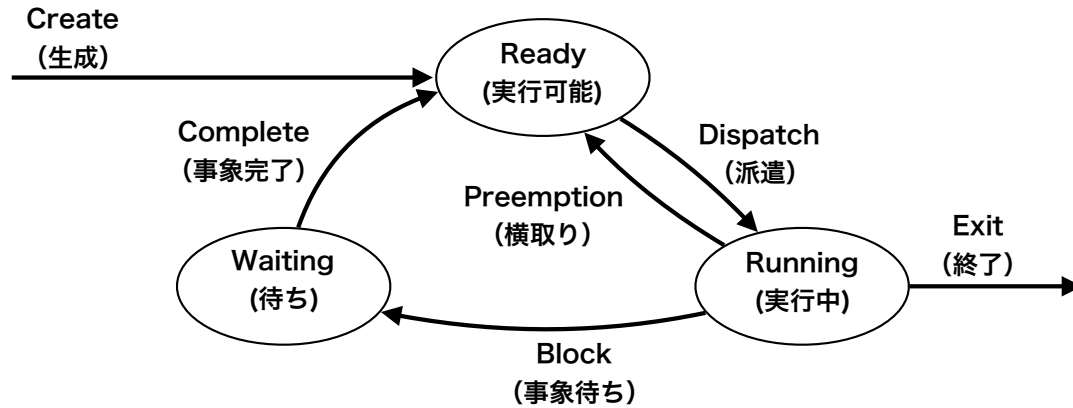
オペレーティングシステム I 令和5年度 後期末試験

(2024.02.14 重村 哲至)

IE4

番 氏名

模範解答



2 実行順序と状態遷移

以下の C 言語風のプログラムにおいて二つの関数 procA() と procB() は、別々のスレッドによって並行実行されます。また、printf() 関数は複数スレッドの環境でも、正常に動作するものとします。

上に示した状態遷移図を参考に以下の問に答えなさい。

```
Semaphore S1 = 1; // 初期値1のセマフォ
Semaphore S2 = 0; // 初期値0のセマフォ

void procA() {
    P( &S1 ); // (1)
    printf("A-1\n"); // (2)
    V( &S2 ); // (3)
    P( &S1 ); // (4)
    printf("A-2\n"); // (5)
    V( &S2 ); // (6)
    printf("A-3\n"); // (7)
}

void procB() {
    P( &S2 ); // (8)
    printf("B-1\n"); // (9)
    V( &S1 ); // (10)
    P( &S2 ); // (11)
    printf("B-2\n"); // (12)
    V( &S1 ); // (13)
}
```

1. 各行を実行中の S1, S2 の値を答えなさい。なお、複数の値の可能性がある場合は全て列挙しなさい。(3 点 × 4 問=12 点)

(2) 行	S1	0
(2) 行	S2	0
(7) 行	S1	0 または 1
(7) 行	S2	0

2. このプログラムの出力には 2 つの可能性があります。両方を答えなさい。

(3 点 × 2 問=6 点)

出力例 1	出力例 2
A-1	A-1
B-1	B-1
A-2	A-2
A-3	B-2
B-2	A-3

3. procA を Block 遷移させる可能性のある全ての行を (1)~(13) の番号で答えなさい。(3 点)
(4)
4. procB を Block 遷移させる可能性のある全ての行を (1)~(13) の番号で答えなさい。(3 点)
(8), (11)
5. procA を Complete 遷移させる可能性のある全ての行を (1)~(13) の番号で答えなさい。(3 点)
(10)
6. procB を Complete 遷移させる可能性のある全ての行を (1)~(13) の番号で答えなさい。(3 点)
(3), (6)

3 相互排除

1. 次の TeC 風のアセンブリ言語で記述した、二つのスレッドで実行されるプログラムの実行結果について答えなさい。

```
// スレッド 1
...
(A) LD GO, NUM
(B) ADD GO, #1
(C) ST GO, NUM
...

// スレッド 2
...
(a) LD GO, NUM
(b) SUB GO, #1
(c) ST GO, NUM
...

// 共有変数
NUM DC 1 // NUMの初期値=1
```

- (a) NUM=1 のとき、 $A \rightarrow B \rightarrow C \rightarrow a \rightarrow b \rightarrow c$ の順で命令を実行した。実行後の NUM の値を答えなさい。(3 点)

NUM の最終値 1

- (b) NUM=1 のとき、 $A \rightarrow B \rightarrow a \rightarrow b \rightarrow C \rightarrow c$ の順で命令を実行した。実行後の NUM の値を答えなさい。(3 点)

NUM の最終値 0

- (c) NUM=1 のとき、 $A \rightarrow a \rightarrow B \rightarrow b \rightarrow c \rightarrow C$ の順で命令を実行した。実行後の NUM の値を答えなさい。(3 点)

NUM の最終値 2

- (d) (A)~(C), (a)~(c) のような部分の名称を答えなさい。(3 点)

(ヒント: 1 の語群にある言葉)

クリティカルセクション

2. 次のような機械語命令が使用できるとき、1. のプログラムの (A)~(C) を INC 命令で、(a)~(c) を DEC 命令で置き換えます。以下の問に答えなさい。

INC (Increment) 命令

INC M

- ① バスをロックする
- ② $T \leftarrow [M]$
- ③ バスのロックを解除する
- ④ $T \leftarrow T + 1$
- ⑤ バスをロックする
- ⑥ $[M] \leftarrow T$
- ⑦ バスのロックを解除する

DEC (Decrement) 命令

DEC M

- ① バスをロックする
- ② $T \leftarrow [M]$
- ③ バスのロックを解除する
- ④ $T \leftarrow T - 1$
- ⑤ バスをロックする
- ⑥ $[M] \leftarrow T$
- ⑦ バスのロックを解除する

- (a) シングルプロセッサシステムでは競合が発生しなくなります。理由を 100 文字以内で説明しなさい。(4 点)

クリティカルセクションが
1 機械語命令で終わる。
単一機械語命令の実行途中で
割り込みは発生しないので、
他のスレッドに切り替わることはなく、
競合は発生しない。

- (b) マルチプロセッサシステムでも競合が発生しないようにするためには、INC, DEC 命令をどのように改良したらよいか 40 文字以内で説明しなさい。(4 点)

命令の途中でバスのロックを解除しないようにすればよい。

4 モニタの応用

次は授業で紹介した仮想言語のモニタを用いてメッセージ通信のリンクを実装した例です。バッファにデータがある時に `send`, または, バッファにデータがない時に `receive` を実行すると, スレッドは条件変数の待ち行列に入ります。メッセージは `int` 型の値を一つだけ含む簡単なものとします。

```
monitor Link {
    int buf;           // メッセージのバッファ
    boolean busy;      // バッファにデータがある
    Condition empty;   // 条件変数
    Condition full;    // 条件変数
    // 初期化プログラム
    Link() {
        busy = false;
    }
    // メッセージ送信
    public void send(int msg) {
        if (busy) full.wait();
        buf = msg;
        busy = ##(a)##;
        ##(b)##;
    }
    // メッセージ受信
    public int receive() {
        if (!busy) ##(c)##;
        int msg = buf;
        busy = ##(d)##;
        ##(e)##;
        return msg;
    }
}

// リンクの使用例
class Main {
    static Link link = new Link();
    public static void main(String[] args) {
        ...
        link.send(msg);           // 送信の例
        ...
        msg = link.receive();      // 受信の例
        ...
    }
}
```

1. プログラム中「##(a)##」に適切な記述を答えなさい。(3点)

true

2. プログラム中「##(b)##」に適切な記述を答えなさい。(3点)

empty.signal()

3. プログラム中「##(c)##」に適切な記述を答えなさい。(3点)

empty.wait()

4. プログラム中「##(d)##」に適切な記述を答えなさい。(3点)

false

5. プログラム中「##(e)##」に適切な記述を答えなさい。(3点)

full.signal()

6. このメッセージ通信機構の特徴に該当する選択肢の言葉に丸を付けなさい。(5点)

項目	選択肢
通信相手の指定方式	直接指定・間接指定
同期方式	同期・非同期
メッセージ形式	タグ付き・タグ無し
メッセージ長	固定長・可変長
バッファリング	あり・なし