オペレーティングシステム I 令和 6 年度 後期末試験

(2025.02.06 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．ただし，(4) は数値を答えなさい．
(1 点 ×25 問＝ 25 点)

ヒント：語群の言葉を全て 1 回使用します．

複数のスレッドが並行実行されるとき，実行のタイミングによって異常な振る舞いをすることがある．このときは (1) が発生している可能性があり (2) を行う必要がある．

(2) が必要なプログラムの部分は (3) と呼ばれる．セマフォを用いて (2) を行う場合は，初期値 (4) のセマフォを用意し，(3) の入口では (5)，出口では (6) を実行する．

セマフォの (5) や (6) は OS 内部のプログラムが提供する．これらは，より低レベルの (2) の仕組みを利用している．この (2) の仕組みとして，シングルプロセッサシステムでは (7) による手法を用いることができる．(7) により，(3) 実行中スレッドが (8) することを防ぐことで目的が達成される．

マルチプロセッサシステムでは，低レベルの (2) に TS 命令などの専用命令を用いる．この方法では，他のスレッドが (3) を脱出するまで (9) により待つので，CPU を無駄遣いしてしまう．

セマフォの (5) や (6) を提供する OS 内部のプログラムは，スレッドの状態を変化させる．(6) は待ちスレッドを (10) 遷移させる可能性がある．(5) は自身スレッドを (11) 遷移させる可能性がある．(ヒント：プロセスの状態遷移図を参照すること)

(12) はメモリ管理のハードウェア (13) を操作し，複数のプロセスに同じ物理メモリを割り付けることで実現できる．使用を開始するためにプロセスはシステムコールを使用する必要があるが，その後はシステムコールを使用することなくプロセス間で情報のやり取りができる．

メッセージ通信には，通信相手の指定にリンクを用いる (14) 方式と，プロセス番号などを用いる (15) 方式がある．メッセージの送受信に用いるシステムコールがブロックすることで，送信プロセスと受信プロセスを (16) させることが可能である．

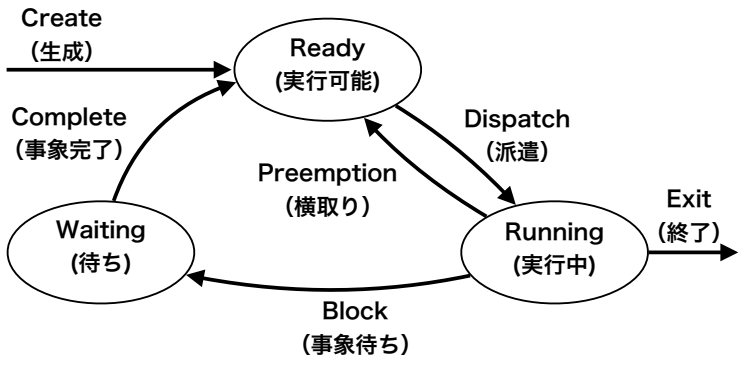
モニタは，リソース管理の機能と制約を持った (17) データ型である．モニタの手続きは (18) の働きによって同時には一つしか実行できない．

モニタの内部には必要に応じて名前付きの (19) を置くことができる．スレッドは (19) に対して，自身が待ち状態になる (20) 操作と，待ち状態のスレッドを実行可能にする (21) 操作をすることができる．

デッドロックの原因となる (22) は，資源の確保順序に制約を付けることで回避できる．しかし，(23) のように順序付けをすると不公平が生じる例もある．

もう一つのデッドロック原因は (24) である．これは複数の資源を同時に確保できる機構があれば回避できる．セマフォ用の同時確保の機構として (25) をあげることができる．

語群：
(あ) Block，(い) Complete，
(う) MMU(Memory Management Unit)，
(え) P_and，(お) P 操作，(か) signal，(き) V 操作，
(く) wait，(け) ガード，(こ) クリティカルセクション，
(さ) ビジーウェイティング，(し) プリエンプション，
(す) 確保待ち，(せ) 間接指定，(そ) 共有メモリ，
(た) 競合，(ち) 食事する哲学者問題，(つ) 循環待ち，
(て) 条件変数，(と) 相互排除，(な) 抽象，
(に) 直接指定，(ぬ) 同期，(ね) 割り込み禁止



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

オペレーティングシステム I 令和 6 年度 後期末試験

(2025.02.06 重村 哲至)　　IE4 ____番　氏名 模範解答

2 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について答えなさい．

```
// スレッド 1
        ...
(A)     LD   G0,NUM
(B)     ADD  G0,#1
(C)     ST   G0,NUM
        ...
// スレッド 2
        ...
(a)     LD   G0,NUM
(b)     SUB  G0,#1
(c)     ST   G0,NUM
        ...
// 共有変数
NUM   DC  1        // NUMの初期値=1
```

(a) NUM=1 のとき，A → a → b → c → B → 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

(b) NUM=1 のとき，A → a → B → b → C → 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 → B → C → a → b → 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

(d) (A) → (B) → (C)，(a) → (b) → (c) はクリティカルセクションと呼ばれます．シングルプロセッサシステムにおいて，クリティカルセクションが必ず連続実行されるようにする，簡単な改良方法を説明しなさい．(3 点)

割り込み禁止状態でクリティカルセクションを実行するようにする．

2. TS(Test and Set) 機械語命令が使用できるとき，シングルプロセッサシステムで下のプログラムを使用した場合について答えなさい．なお，TS 命令はメモリの値がゼロのときゼロフラグを 1 にするものとする．

```
// エントリーセクション
(A) L1    DI           // 割り込み禁止
(B)       TS   G0,FLG  // FLGが0ならZ←1
(C)       JZ   L2
(D)       EI           // 割り込み許可
(E)       JMP  L1
// クリティカルセクション
    L2    ...
// エグジットセクション
(1)       LD   G0,#0
(2)       ST   G0,FLG
(3)       EI           // 割り込み許可
// エグジットセクション終了
        ...
// 共有変数
FLG     DC  0
```

(a) クリティカルセクションの実行が排他的にされるか説明しなさい．(4 点)

シングルプロセッサシステムならクリティカルセクションを割り込み禁止で実行すれば相互排除がされる．(A)〜(3) まで割り込み禁止になっておりクリティカルセクションはその中に含まれる．よって，クリティカルセクションは割り込み禁止で排他的に実行される．

(b) エントリーセクションでビジーウェイティングが発生する可能性について説明しなさい．(4 点)

ビジーウェイティングするのは FLG が 1 でエントリーセクションに来た場合だけである．(A)〜(3) まで割り込み禁止になっているので，FLG が 1 のときプリエンプションすることはない．よって，ビジーウェイティングは発生しない．

オペレーティングシステム I 令和 6 年度 後期末試験

(2025.02.06 重村 哲至)　　IE4 ____番　氏名 模範解答

3　セマフォ

1. 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      data;     // データ
Semaphore S1;       // データの相互排除用
void writerThread() {
  for ( ; ; ) {
    Data d = produce(); // データを作る
    _(a)_;
    writeRecores( d );  // writeする
    _(b)_;
  }
}
int       cnt = 0;    // リーダ数
Semaphore S2;         // cntの相互排除用
void readerThread() {
  for ( ; ; ) {
    _(c)_;
    if ( cnt == 0 ) _(d)_;
    cnt = cnt + 1;
    _(e)_;
    Data d = readRecords(); // readする
    _(f)_;
    cnt = cnt - 1;
    if ( cnt == 0 ) _(g)_;
    _(h)_;
    consume( d );     // データを使う
  }
}
```

(a) 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．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．(2 点 ×8 問=16 点)

(a)	P(S1)
(b)	V(S1)
(c)	P(S2)
(d)	P(S1)
(e)	V(S2)
(f)	P(S2)
(g)	V(S1)
(h)	V(S2)

(b) 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．
(3 点 ×2 問=6 点)

S1	1
S2	1

2. 次は C 言語風の言語を用いて相互排除の例を記述したものです．複数のスレッドが変数 num の値を正しく更新できるようにセマフォを用います．

```
int       num = 1;
Semaphore S;
void inc() {
  _(a)_;
  num++;
  _(b)_;
}
void dec() {
  _(c)_;
  num--;
  _(d)_;
}
```

(a) セマフォ（S）の初期値はいくつにするべきか答えなさい．(3 点)

1

(b) 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)

オペレーティングシステムⅠ 令和6年度 後期末試験

(2025.02.06 重村 哲至)　　IE4　　番　氏名　模範解答

4　モニタ

次は Java 風の仮想言語で記述したセマフォの実装例です．cnt は，モニタで待ち状態になったスレッドの数を管理します．

```
int val;          // セマフォの値
Condition c;      // 条件変数
int cnt;          // 待ちスレッドの数

// 初期化プログラム
Semaphore(int n) {
  val = n;
  cnt = 0;
}

// P操作
public void P() { // ガードに守られる
  if (_(a)_) {
    val--;
  } else {
    cnt++;
    c.wait();
    _(b)_;
  }
}

// V操作
public void V() { // ガードに守られる
  if (_(c)_) {
    val++;
  } else {
    c.signal();
  }
}
}
```

1. プログラム中 (a)，(b)，(c) に適切な記述を答えなさい．(3 点 ×3 問=9 点)

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

2. 次は Java 風の仮想言語で記述したモニタの使用例です．procA()，procB() メソッドは，別々のスレッドによって並行実行されるものとします．また，複数スレッドの環境でも正常に動作する C 言語風の printf() 関数が使用できるものとします．

```
Semaphore S1 = new Semaphore(0);
Semaphore S2 = new Semaphore(1);

void procA() {
  S1.P();
  printf("A-1\n");
  S2.V();
  S1.P();
  printf("A-2\n");
}

void procB() {
  S2.P();
  printf("B-1\n");
  S1.V();
  S2.P();
  printf("B-2\n");
  S1.V();
}
```

(a) このプログラムの出力を答えなさい．(5 点)

B-1
A-1
B-2
A-2

(b) プログラム実行終了時のセマフォの値を答えなさい．(2 点 ×2 問=4 点)

S1	0
S2	0