OS1 H28 年度 後期末試験　　(2017.02.03 重村 哲至)

IE4　　番　氏名　模範解答

1. 次の文章の空欄に適切な言葉を、語群から記号で答えなさい。(2 点× 15 問= 30 点)

プロセス間の相互干渉の種類は、知らないうちに他のプロセスから受ける **(1)** 、複数のプロセスが情報交換を行いながら協力して処理を進める **(2)** 、資源を取り合う **(3)** がある。**(3)** が発生する場合は正しく処理ができるように資源のアクセスを同時には一つのプログラムしか行わないように **(4)** 制御する必要がある。また、プログラム中で **(4)** 制御が必要な部分を **(5)** と呼ぶ。

セマフォは **(4)** 制御に用いられる機構である。セマフォにはプロセスを「待ち」にすることがある **(6)** と「待ち」のプロセスを「実行可能」にするこがある **(7)** が使用できる。セマフォの種類は、0, 1 の二つの値しか取らない **(8)** セマフォと **(9)** セマフォがある。

セマフォとは異なる **(4)** 制御機構として、整理券に似た考え方を用いるものがある。この方式では、整理券発行器に当たる **(10)** とサービス中の整理券番号を表示する機械に当たる **(11)** を用いる。

プロセス間のメッセージ通信機構は、通信相手の指定方式により二つに分類される。一つは通信先として **(12)** を指定する直接方式である。もう一つは通信先として **(13)** を指定する間接方式である。

「食事する哲学者の問題」では、哲学者が片手にフォークを持ったまま、もう一方の手に持つフォークが使用できるまで待つことがある。このような待ちを **(14)** 待ちと呼ぶ。また、全ての哲学者が同じ側の手にフォークを持ったまま待ちになる可能性がある。この時は **(15)** 待ちが発生している。

語群：

（あ） 2 進（バイナリ）、（い） P 命令、（う） V 命令、（え） イベントカウント、
（お） クリティカルセクション、（か） シーケンサ、（き） プロセス、（く） リンク、
（け） 確保、（こ） 干渉、（さ） 競合、（し） 協調、（す） 計数（カウンティング）、
（せ） 排他、（そ） 循環

(1)	（こ）	(2)	（し）	(3)	（さ）	(4)	（せ）	(5)	（お）
(6)	（い）	(7)	（う）	(8)	（あ）	(9)	（す）	(10)	（か）
(11)	（え）	(12)	（き）	(13)	（く）	(14)	（け）	(15)	（そ）

OS1 H28 年度 後期末試験　(2017.02.03 重村 哲至)

IE4 ____番　氏名 模範解答

2. 次の TaC 風のアセンブリ言語プログラムは、スピンロック（ビジーウェイティング）による排他制御機構です。SWAP 命令を用いマルチプロセッサでも使用できるエントリーシーケンスを下の指示に従い完成しなさい。なお、SWAP 命令の仕様はプログラムの下に示した通りです。(10 点)

```
; エントリーシーケンス
L1    DI                ; 割込み禁止
      ##(a)##
      EI                ; 割込み許可
      JMP  L1
L2
; クリティカルセクション
      ...
; イクシットシーケンス
      LD   G0, #0
      ST   G0, FLG
      EI
      ...
FLG   DC   0     ; 初期値 1 のフラグ

;SWAP 命令
; 書式：SWAP G0, FLG
; 意味：レジスタの値とメモリの
;         値を入れ替える。その間、他の
;         プロセッサはメモリにアクセス
;         できない。
```

上のプログラム中##(a)##に適切な数行のプログラムを以下に書きなさい。

```
LD   G0, #1
SWAP G0, FLG
CMP  G0, #0
JZ   L2
```

3. 次の C 言語風のプログラムはセマフォを用いた排他制御の例です。プロセス A とプロセス B が同じ資源をアクセスする場合、空欄に適切な記述を答えなさい。
(3 点× 5 問=15 点)

```
// プロセス間で共有される
// セマフォS（適切な初期値を答える）
SEMAPHORE S = ##(A)##;

// プロセス A が以下を実行
processA() {
   ...
   ##(B)##;
   クリティカルセクション
   ##(C)##;
   ...
}

// プロセス B が以下を実行
processB() {
   ...
   ##(D)##;
   クリティカルセクション
   ##(E)##;
   ...
}
```

(A)	1
(B)	P(S)
(C)	V(S)
(D)	P(S)
(E)	V(S)

4. 次の C 言語風のプログラムは、単一プロデューサ／単一コンシューマ問題のセマフォによる解です。空欄に適切な記述を答えなさい。(3 点× 5 問=15 点)

```
#define N 10

// 以下はプロセス間で共有されるもの
SEMAPHORE S = ##(A)##;
SEMAPHORE M = 0;  // 初期値ゼロ
MESSAGE BUF[N];   // メッセージ配列

// 生産者プロセスが以下を実行
producer() {
  int i=0;
  while (true) {
    メッセージを作る;
    ##(B)##;
    BUF[i]=メッセージ;
    ##(C)##;
    i=(i+1)%N;
  }
}

// 消費者プロセスが以下を実行
consumer() {
  int j=0;
  while (true) {
    ##(D)##;
    メッセージ=BUF[j];
    ##(E)##;
    メッセージを使用する;
    j=(j+1)%N;
  }
}
```

(A)	N	(B)	P(S)
(C)	V(M)	(D)	P(M)
(E)	V(S)		

5. 次の C 言語風のプログラムは、単一プロデューサ／単一コンシューマ問題のイベントカウントによる解です。空欄に適切な記述を答えなさい。(3 点× 5 問=15 点)

```
#define N 5

// 以下はプロセス間で共有されるもの
EVENTCOUNT IN  = 0; // イベント
EVENTCOUNT OUT = 0; //   カウント
MESSAGE BUF[N];   // メッセージ配列

// 生産者プロセスが以下を実行
producer() {
  int i=0;
  while (true) {
    メッセージを作る;
    await(OUT,##(A)##);
    BUF[i % N]=メッセージ;
    advance(##(B)##);
    i++;
  }
}

// 消費者プロセスが以下を実行
consumer() {
  int j=0;
  while (true) {
    await(##(C)##, ##(D)##);
    メッセージ=BUF[j % N];
    advance(##(E)##);
    メッセージを使用する;
    j++;
  }
}
```

(A)	i-N+1	(B)	IN
(C)	IN	(D)	j+1
(E)	OUT		

6. 次の C 言語風のプログラムは、リーダ・ライタ問題のシーケンサとイベントカウントによる解です。よく読んで下の問に答えなさい。(5 点× 3 問=15 点)

```
SEQUENCER  W = 0;
EVENTCOUNT X = 0;

// ライタプロセス
writer() {
  int w;
  while (true) {
    w=ticket(W);
    await(X, w);
    データ書き込み ();
    advance(X);
  }
}

SEQUENCER  M = 0;
EVENTCOUNT Y = 0;
int        R = 0;

// リーダプロセス
reader() {
  int m, r;
  while (true) {
    m=ticket(M);
    await(Y, m);
    if (R==0) {
      r=ticket(W);
      await(X, r);
    }
    R++;
    advance(Y);
    データ読み出し ();
    m=ticket(M);
    await(Y, m);
    R--;
    if (R==0) advance(X);
    advancd(Y);
}
```

(a) 変数 R は何を表す変数か答えなさい。

データを読み出し中のリーダプロセスの数を表している。

(b) シーケンサ M とイベントカウント Y は、何の目的で使用されているか答えなさい。

複数のリーダが変数 R をアクセスする順序を決めるために使用されている。

(c) シーケンサ W とイベントカウント X は、何の目的で使用されているか答えなさい。

ライタプロセスとリーダプロセスでデータにアクセスする順序を決めるために使用されている。