OS 第４回予習確認テスト　　学生番号（　　）　　氏名（　　）

（グループ内で交換して採点）　　（合計　　点　／　９点）

1） 次の文の空欄を埋めよ。（0．２×13＝2．６点）

１台しかないプリンタを2人以上のユーザで共有する場合、出力が交じり合わないためには （　　） が必要になる。 この仕組を使うと、プロセス A によって使用中の場合（　　）することによって、他のプロセス B が使用しようとすると（　　）される。

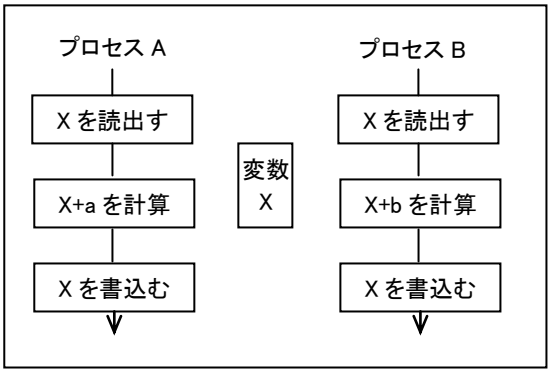
実際の実現の仕組として、（　　）とプロセスの状態遷移による方法がある。前者は、値が1なら使用中、0なら空きを示す変数xを共有し、（　　）するときには1を書込み、（　　）するときには0を書込む。資源を利用したいときにはこの共有変数xの値をチェックし、もしxが1になっていれば使用中なので、再び（　　）し0になるまでこれを繰り返す。 このため、（　　）という欠点がある。

一方、プロセスの状態遷移による方法は、繰り返してチェックをするループを回ることをやめ、プロセスを（　　）状態に遷移させる。 現在資源を使用中のプロセスは、（　　）操作をするときに、待っているプロセスの状態を（　　）状態に遷移させる。これによって、待っていたプロセスは次の機会にスケジュールされる。

この2つの方法を比較すると、前者の方法は（　　）の問題があるがそれは状態遷移の方法では起こらない。他方、状態遷移による方法は前者の方法に比べて（　　）という欠点がある。

2） 次の問に答えよ。（0．８×２＝１．６点）

前問で議論した1）と同じ種類の問題として、共有されている変数 X を2つのプロセス A と B が同時に更新する場合が考えられる。 A、B が右図にあるようなプログラムである場合、A と B がともに実行されれば X の値は a+b となるはずであるが、タイミングによってそうならないことがある。



（あ） X の値が a+b とはならない場合の動作順序（タイミング）を説明せよ

（い） タイミングに依らず期待した動作をするためには、どうすればよいか （注： X+a をやってから+b をしても、逆の順序で X+b をやってから+a をしても、結果は変わらないことにも注意して欲しい。）

3） 問2）で見た動作順序（タイミング）の制御で、ロックして排他制御しようとする（クリティカルリージョンを実現する）とき、どうやったらできるか次の順序で考えてみたい。（2点）

① 問1）で見たように「使用中」を示す変数L（使用中ならL＝1、そうでないならL＝0）を共有変数として用意し、それに従ってビジーウェイトによりロックすれば、排他制御が実現できるだろう。まずその原理の図を描いてみよう。（0．６点）

② この使用中を示す共有変数Lの値の変更の様子を考えてみる。まずLを読んでみて、読んだ結果の値が0ならば他プロセスが使っていないので、自分が使うことができる。だから、まずLに1を書き込んで（他プロセスに対して）「使用中」状態を宣言して、それから自分の仕事をする。このLを「読みだして（判断して）書く」操作は、問2）と同じ形をしていて、同じようなタイミング上のトラブルを生じる可能性がある。 問2の図と同じように、思ったように行かないタイミングを描いて、何が起こるか説明してみよう。（0．7点）

③ このうまくいかないタイミングの問題を解決するには、どうしたらよいか、説明せよ。（0．7点）

4） 次の文の空欄を埋めよ。（0．2×14＝2．8点）

並行処理のタイミングに関するもう1つの問題として、たとえばバッファへの書き込みと読み出しの前後関係の制御がある。 この制御をプロセスの（　　）と呼ぶ。 やりたい制御をもう少し詳しく説明するならば、情報生産者であるプロセス A がバッファ S に書き込み、それを消費者であるプロセス B が読み出して利用する。

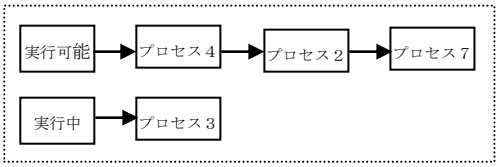
このとき、もし、（　　が　　）前に（　　が　　）と、正しいデータを受け取ることができない。また逆に、（　　が　　）前に（　　が　　）と、データが上書きされ、上書きされたデータは B に渡ることなく消滅してしまう。 だから、A と B は順番を守って交互にバッファの書き込み・読み出しをしなければならない。

この例の場合、プロセス（　　）はプロセス（　　）が書き込むのを待って読み出しを行なうように制御し、またプロセス（　　）は、プロセス（　　）が読み出すのを待って次の書き込みを行うように制御する。

プロセスの同期をするために用いられる（ハイレベルの）仕組みとして、（計数）セマフォがある。（計数）セマフォには、ロック操作（自分が排他的な資源を使いたいときに、空いていなければ待たされる）に相当する（　　）操作と、アンロック操作（資源を使い終わったので解放する）に相当する（　　）操作がある。（計数）セマフォが単純なロック操作と異なる点は、共有のセマフォ変数 s をもっており、s の初期値をある整数 N に設定すると、（（何ができる）　　）が、（（それを越えるとどうなる）　　）。 なお、初期値を1とした場合はバイナリ・セマフォと呼ばれ、（　　）と同等の動作になる。

5）（時間があれば） 複合データの統一性（インテグリティ）の維持（点数外）

たとえば、OS の中ではプロセスを管理するデータをリンクリストの形で持っている。リンクリストの途中のノードをリストから外して別のリストに入れる場合、他のプロセスが同じリストを操作しようとすると、



① どういう問題が起こるか

② それを防ぐにはどうすればよいか