

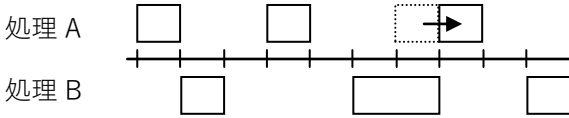
OS 中間試験コメント

[1] ① どのような仕組みによって、プロセッサの仮想化を実現するのか。

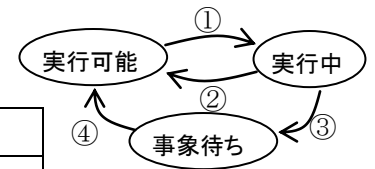
プログラムを切り替えることによって、複数のプログラムをあたかも同時並行に実行しているように見せる。並行に実行するプログラム（プロセス）はそれぞれが仮想的なプロセッサ上で実行されているように見える。

② どのように実行されるかをタイムチャート上に描け。（5点）

処理 A の CPU と B の CPU が重なってはいけない。 ⇒ 減点



[2] ① 状態遷移図を描け



②

状態の名前	どのような状況の時の状態かの説明
実行可能	CPU が空きさえすれば、すぐに実行を開始できる状態。空き待ち。
実行中	CPU が割り当てられ、プログラムが実行している状態。
事象待ち	CPU 外部で発生する事象を待っている状態。

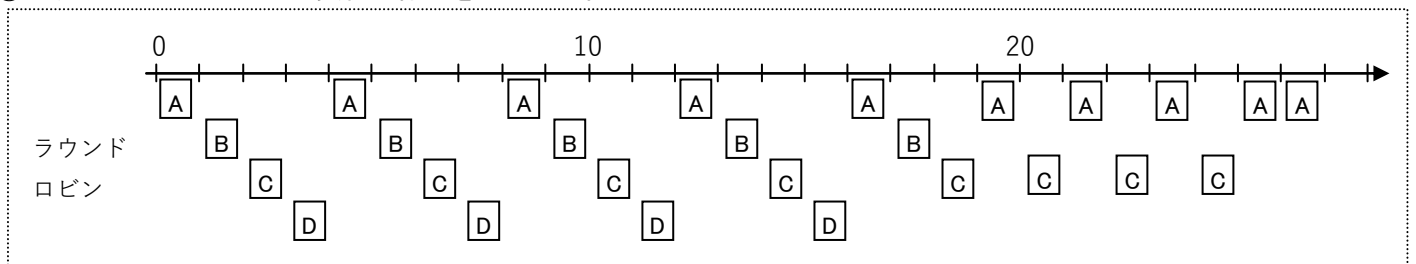
③ 状態遷移の説明

	状態遷移が起こるきっかけ
(1)	CPU が空いたので CPU が割り当てられた(ディスパッチされた)
(2)	優先度が高いプロセスが実行可能になり、もしくは与えられた時間(タイムスライス)が尽きたため自分は CPU をとり上げられた(プリエンブションされた)
(3)	実行中に、外部事象を待つような操作(例えば入出力装置を起動したり、目覚まし時計をセットしたり)して、事象の発生を待つ状態に移る。
(4)	待っていた事象が発生したので、その後の処理を再開する

[3] ① プロセスの実行スケジューリングとは何か 複数ある実行可能プロセスから次に CPU を割り当てるプロセスを選ぶ

③ FCFS と SPTF を比較せよ 平均待ち時間(時刻 0 からプロセス開始までの時間の平均)で比較すると、SPTF が短いので、ユーザにとって開始までの待ち時間が短くなる。

④ ラウンドロビン方式の実行の様子を図示せよ。



⑤ 連続して実行する方式とラウンドロビンについて、比較せよ。

ラウンドロビンは、先に実行されているプロセスが終了するのを待つことがない。1周待てば実行される。

(ユーザが端末で)応答を待つ時間(ターンアラウンド時間)が短い。

[4] Windows や Linux は (ハード) リアルタイム性がないと言われるが、どのような機能が欠けているためか。

ラウンドロビンの場合、次に CPU が割り当てられるまでの時間は1周する時間になるので、その1周に含まれる他のプロセスの数(つまり実行可能状態にあるプロセスの総数)によって定まる。1周の時間に上限を設ける(ハードリアルタイム)ためには、実行可能状態になるプロセスの総数の上限を限る必要があるが、これを限る機能が無いのでハードリアルタイムでない。

[5] ① 相互排除とは何か。 資源を複数のプロセスが共有するとき、1時点で利用できるプロセスの数が1つに制限されていること。

② (1) この現象が起こる理由を説明せよ。 Aが共有変数を読んで「空き」0と判断してから、自分が使用中1であることを書込むまでの間に、時間があるために、その間にBが共有変数を読んで「空き」0を得てしまうから。

(2) Aが共有変数を読んでから書くまでの間に、他のプロセスが読み出すことができないようにする。

(余談) このために、共有変数をロックすればよいと思うのだが、今「ロック」を実現しようとしているのであり、「ロック」を実現するための道具として中で「ロック」を使うのだと、またその「ロック」のために中で「ロック」を使い、というわけで、無限に続いてしまう。これは具合が悪い。どうするかというと、(変数を置く)メモリに「読んで書く、を不可分に行う」という操作を設ける。いくつかのパターンが提案されているが、たとえば「Test and Set 命令」がある。これはハードウェア的に、テストしてからセットが終わるまで他の操作ができないようになっている。

③ ビジーウェイトとプロセス状態遷移による方法とを比較せよ

比較のポイント	ビジーウェイト方式	プロセス状態遷移による方法
CPUの無駄	待っている間はずっとCPUを使い続けるのでその間無駄になる	CPUを放棄して事象待ち状態になるので、無駄遣いはない
切替え時間(応答)	常時見張っているので、ロックが解除されればすぐに取れる	プロセスの状態の切り替えには時間がかかる

[6] デッドロックについて次の問に従って説明してください。(13点+加点5点)

① 略

② デッドロックの不都合 デッドロックを起こしているプロセスは、事象待ち状態のまま処理が先に進まない。

③ 必要十分条件がどういう意味か説明せよ

(1) 相互排除 資源が1つのプロセスしかアクセスできない。

(2) 確保待ち 資源をアクセスするとき、確保できなければ(誰かがロックしていれば)空くまで待つ。

(3) 横取り不可 使用中の資源を、別のプロセスが横取りすることはできない。

(4) 循環待ち 資源の待ちが循環している。互いに待ちあっている。3すくみ状態である。..

④ デッドロックの対処法を説明せよ

(1) デッドロックの発生後に、資源グラフに循環があることによって検出し、どれか1つのプロセスの処理を、(循環している)資源の利用開始前の状態まで戻す。(ロールバック) ..

(2) この資源要求によってデッドロックになる、という要求を資源グラフによって検出し、その要求を出したプロセスを、(循環している)資源の利用開始前の状態まで戻す。(ロールバック)

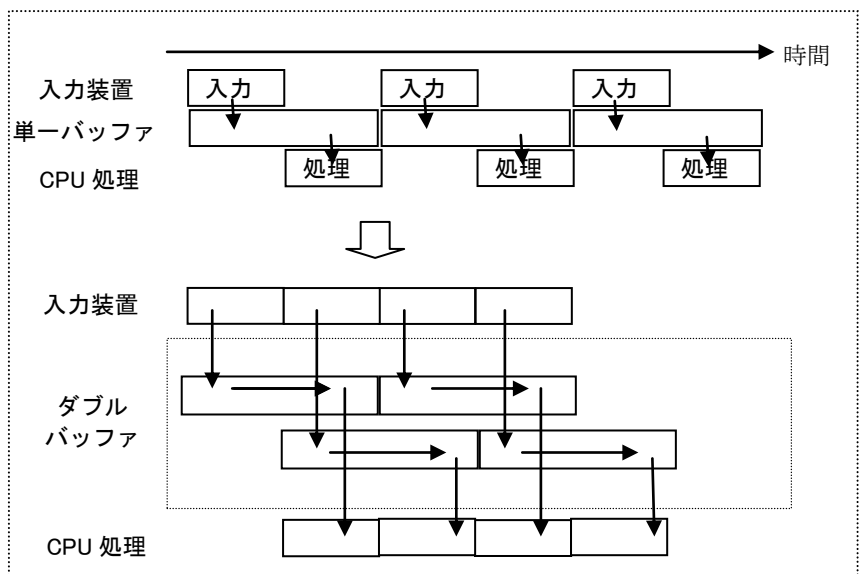
(3) デッドロックが発生しないようにプログラムを書く。具体的には、(A)資源を一度にロックする、(B)資源のロックに順番を付ける、などの方法がある。

[7] デバイスの仮想化の考え方を説明せよ

異なる種類の入出力デバイスを、おなじやり方(インターフェース)で使えるようにすること。同じプログラムで(プログラムを書き換えなくても)、デバイスを利用できるようになる。

[8] ① 右図の下の方に、ダブルバッファでの動作を書き足せ

② 単一バッファとダブルバッファの動作の違い



いを説明せよ。

単一バッファだと、入力装置は CPU の処理が済んでバッファが空くまで次の入力をできないが、ダブルバッファだと、次の入力を別のバッファに(同時に)書き込むことができる。

### ③ ダブルバッファのメリット

バッファが空くの待たなくてよくなるので、入力の速度が速くなる。

[9] プリフェッチの効果を説明せよ。

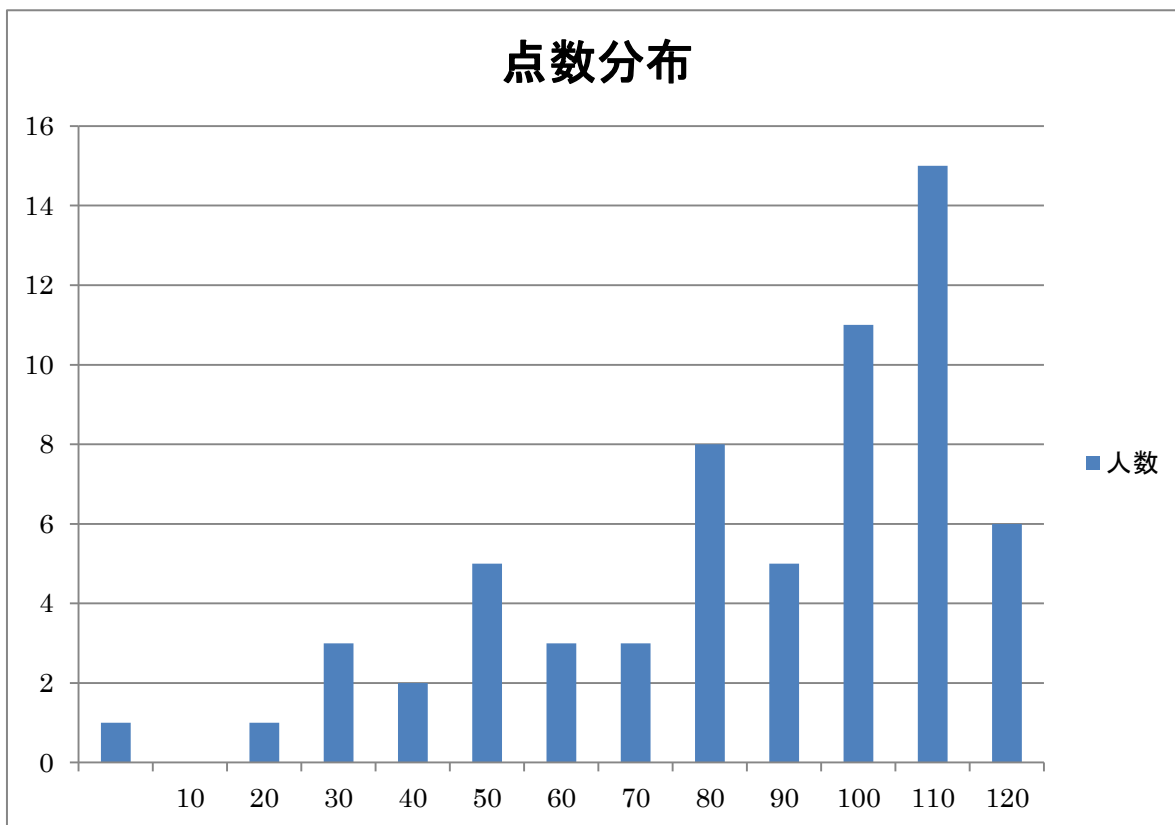
#### ① プリフェッチとはどういうことか

入出力において、今読んでいるデータより1つ先のデータを先読みしておくこと。

#### ② プリフェッチは、(A) どのようなときに有効なのか、(B) どのようなときに無効なのか、説明せよ。

(A) 有効なとき 必ず入力媒体の順序通りに読むことが分かっているとき。たとえば、音楽・ビデオはほとんどの場合先頭から順番通りに再生するので、有効である。

(B) 無効なとき ランダムな順序で読むとき。ファイル内のあちこちのデータをとびとびに読むとき。



かなり甘く(およそ正しければマルにした)採点したからか、全体に出来が良かった。

80点未満の人は、もう少し勉強して欲しい。

素点に0.3を掛けて、他の点数(平常点、期末試験)と足す予定。