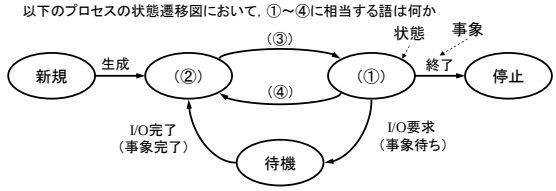
第15回　総復習と期末試験

期末試験では電卓のみ持ち込み可
（教科書，プリント，PCなどは持ち込み不可）

プロセスの状態遷移

以下のプロセスの状態遷移図において，①～④に相当する語は何か

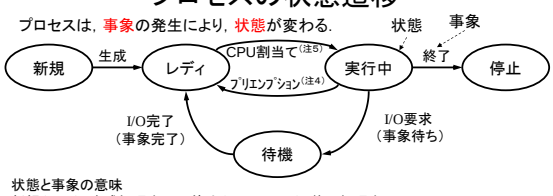


①＿＿＿＿＿②＿＿＿＿＿③＿＿＿＿＿④＿＿＿＿＿.

第7回のスライド

プロセスの状態遷移

プロセスは，事象の発生により，状態が変わる．

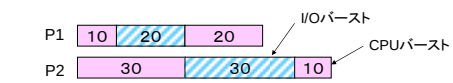


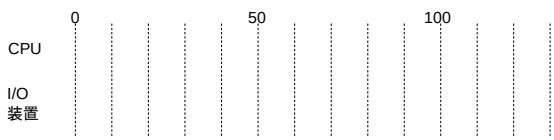
状態と事象の意味
新規(new)：生成処理中　　停止(terminated)：終了処理中
レディ(ready)：実行の準備ができている．(実行可能状態，実行待ち状態とも言う)
実行中(running)(注1)：命令を実行している．
　　ﾌﾟﾛｾｽの定義：実行中のﾌﾟﾛｸﾞﾗﾑの「実行中(in execution)」とは意味が違う
待機(waiting)：I/O処理などで命令が実行できない．(待ち状態，待機中状態とも言う)

生成：プロセスの生成処理の完了　　終了：実行の終了(最後の命令を実行)
CPU割当て：OSがレディ状態のプロセスを選び，CPUを割り当てる
I/O要求：実行中状態のプロセスが，I/O処理を要求するシステムコールを発行
I/O完了：プロセスが要求したI/O処理(ハードウェアの処理)が完了
ﾌﾟﾘｴﾝﾌﾟｼｮﾝ：OSが実行中状態のプロセスからCPUを取りあげ，実行を中断させる

プロセスの実行と状態（順次実行）

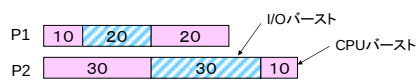
以下のプロセスをP1、P2の順に、順次実行させた場合、開始から終了までのCPU使用率[%]を求めよ。

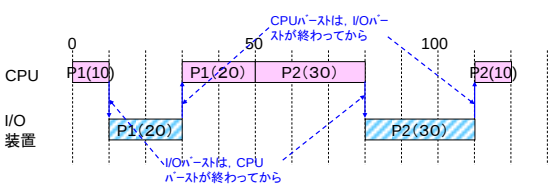




プロセスの実行と状態（作図のポイント）

以下のプロセスをP1、P2の順に、順次実行させた場合、開始から終了までのCPU使用率[%]を求めよ。

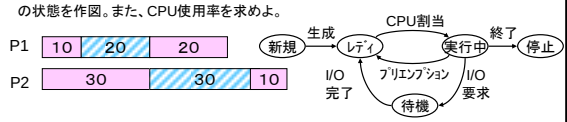


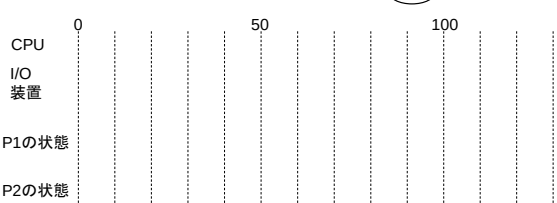


CPU使用時間：10+20+30+10=70(CPUﾊﾞｰｽﾄの合計)　　実行時間120
CPU使用率=70/120=0.583　　58.3[%]
作図しなくても簡単に求められる．

プロセスの実行と状態（多重実行1－1）

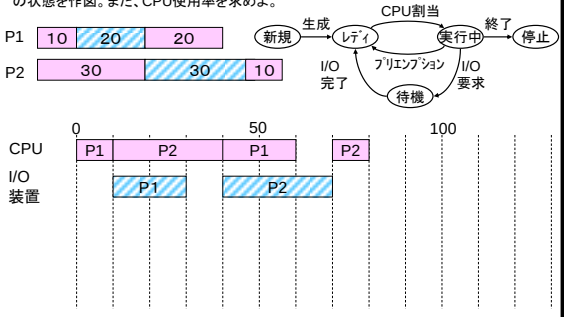
P1、P2を多重し、P1を先に実行させた場合にCPU、I/O装置の使用状況と各プロセスの状態を作図。また、CPU使用率を求めよ。





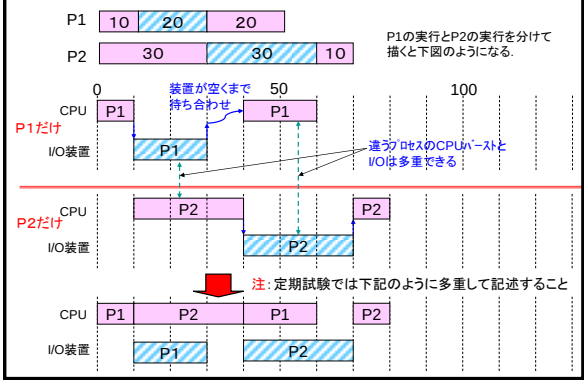
プロセスの実行と状態（多重実行1－1）

P1、P2を多重し、P1を先に実行させた場合にCPU、I/O装置の使用状況と各プロセスの状態を作図。また、CPU使用率を求めよ。



CPU使用率＝70／80＝0．875＝87．5%

作図のポイント（説明図）

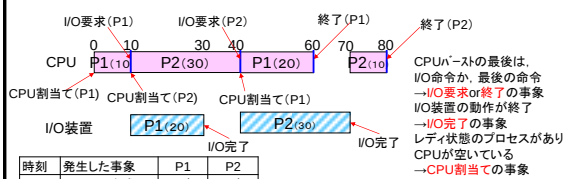


プロセスの実行と状態（多重実行1－2）

前問の実行において，下表の事象が発生した。表の空欄に時刻と状態名を記入せよ。

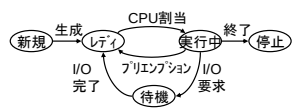
時刻	発生した事象	P1	P2
—	（P1,P2生成）	レディ	レディ
	CPU割当て（P1）		
	I/O要求 CPU割当て		
	I/O完了		
	I/O要求 CPU割当て		
	終了		
	I/O完了 CPU割当て		
	終了		

プロセスの実行と状態（多重実行1－2）



CPUバーストの最後は，I/O命令か，最後の命令
→I/O要求or終了の事象
I/O装置の動作が終了
→I/O完了の事象
レディ状態のプロセスがあり
CPUが空いている
→CPU割当ての事象

時刻	発生した事象	P1	P2
—	（P1，P2生成）	レディ	レディ
0	CPU割当て（P1）		
10	I/O要求 CPU割当て		
30	I/O完了		
40	I/O要求 CPU割当て		
60	終了		
70	I/O完了 CPU割当て		
80	終了		

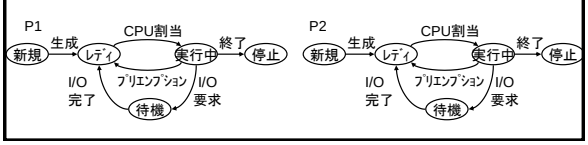


プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	（P1，P2生成）	レディ	レディ
0	CPU割当て（P1）		
10	I/O要求 CPU割当て		
30	I/O完了		
40	I/O要求 CPU割当て		
60	終了		
70	I/O完了 CPU割当て		
80	終了		



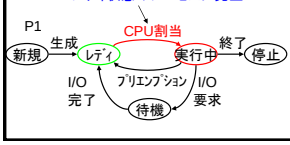
プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

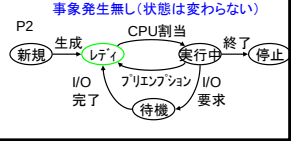
これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	（P1，P2生成）	レディ	レディ
0	CPU割当て（P1）	実行中	レディ
10	I/O要求 CPU割当て		
30	I/O完了		
40	I/O要求 CPU割当て		
60	終了		
70	I/O完了 CPU割当て		
80	終了		

レディ状態のプロセスに発生



事象発生無し（状態は変わらない）

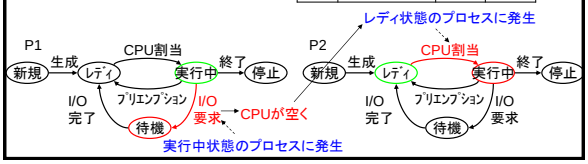


プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	(P1, P2生成)	レディ	レディ
0	CPU割当て(P1)	実行中	レディ
10	I/O要求	待機	レディ
	CPU割当て	待機	実行中
30	I/O完了		
40	I/O要求		
	CPU割当て		
60	終了		
70	I/O完了		
	CPU割当て		
80	終了		

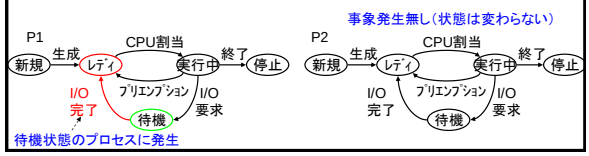


プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	(P1, P2生成)	レディ	レディ
0	CPU割当て(P1)	実行中	レディ
10	I/O要求	待機	レディ
	CPU割当て	待機	実行中
30	I/O完了	レディ	実行中
40	I/O要求		
	CPU割当て		
60	終了		
70	I/O完了		
	CPU割当て		
80	終了		

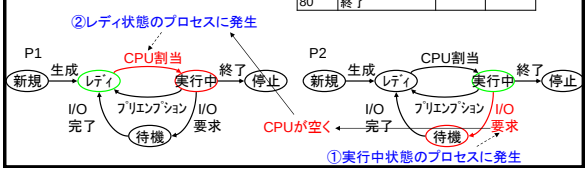


プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	(P1, P2生成)	レディ	レディ
0	CPU割当て(P1)	実行中	レディ
10	I/O要求	待機	レディ
	CPU割当て	待機	実行中
30	I/O完了	レディ	実行中
40	I/O要求	レディ	待機
	CPU割当て	実行中	待機
60	終了		
70	I/O完了		
	CPU割当て		
80	終了		

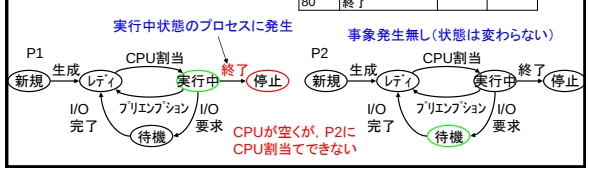


プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	(P1, P2生成)	レディ	レディ
0	CPU割当て(P1)	実行中	レディ
10	I/O要求	待機	レディ
	CPU割当て	待機	実行中
30	I/O完了	レディ	実行中
40	I/O要求	レディ	待機
	CPU割当て	実行中	待機
60	終了	停止	待機
70	I/O完了		
	CPU割当て		
80	終了		

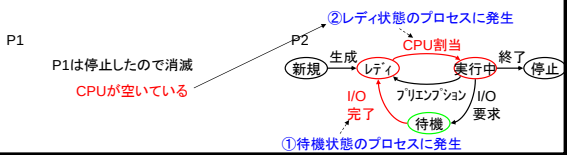


プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	(P1, P2生成)	レディ	レディ
0	CPU割当て(P1)	実行中	レディ
10	I/O要求	待機	レディ
	CPU割当て	待機	実行中
30	I/O完了	レディ	実行中
40	I/O要求	レディ	待機
	CPU割当て	実行中	待機
60	終了	停止	待機
70	I/O完了		レディ
	CPU割当て		実行中
80	終了		

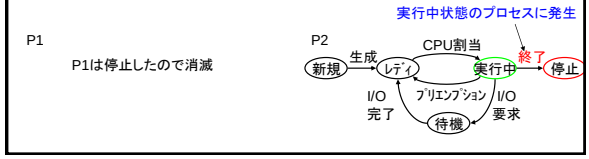


プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

時刻	発生した事象	P1	P2
—	(P1, P2生成)	レディ	レディ
0	CPU割当て(P1)	実行中	レディ
10	I/O要求	待機	レディ
	CPU割当て	待機	実行中
30	I/O完了	レディ	実行中
40	I/O要求	レディ	待機
	CPU割当て	実行中	待機
60	終了	停止	待機
70	I/O完了		レディ
	CPU割当て		実行中
80	終了		停止



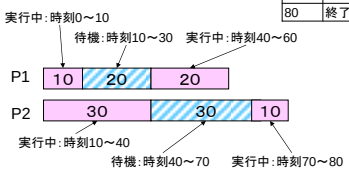
プロセスの実行と状態（多重実行2）

生成が完了している2つのﾌﾟﾛｾｽがある．

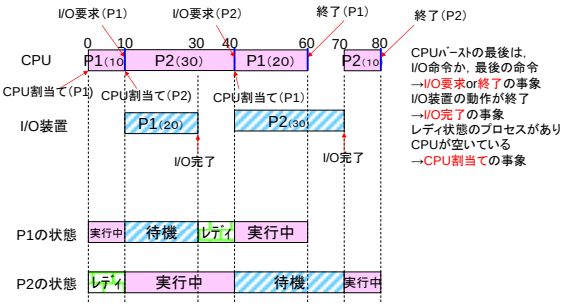
これを並行して実行すると，右表の事象が発生した．空欄に事象発生後の各プロセスの状態を記入せよ．

プロセス単独の実行は右の表により以下のようになる

時刻	発生した事象	P1	P2
—	(P1, P2生成)	レディ	レディ
0	CPU割当て(P1)	実行中	レディ
10	I/O要求	待機	レディ
	CPU割当て	待機	実行中
30	I/O完了	レディ	実行中
40	I/O要求	レディ	待機
	CPU割当て	実行中	待機
60	終了	停止	待機
70	I/O完了		レディ
	CPU割当て		実行中
80	終了		停止



プロセスの実行と状態（多重実行2）



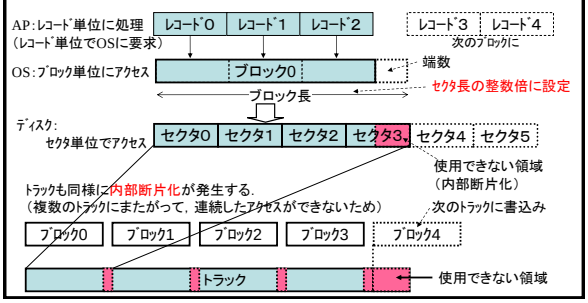
CPUﾊﾞｰｽﾄの最後は，I/O命令か，最後の命令
→I/O要求or終了の事象
I/O装置の動作が終了
→I/O完了の事象
レディ状態のプロセスがありCPUが空いている
→CPU割当ての事象

CPU使用中→実行中，I/O装置使用中→待機
I/O完了～CPU使用までの間→レディ

第4回のスライド

ﾃﾞｨｽｸｱｸｾｽ（ﾚｺｰﾄﾞ，ﾌﾞﾛｯｸ，ｾｸﾀ，ﾄﾗｯｸの関係）

APはﾚｺｰﾄﾞ単位に処理（例えば学生一人分のデータなど）．OSへの要求もﾚｺｰﾄﾞ単位．
OSは効率化のために，複数ﾚｺｰﾄﾞを1ﾌﾞﾛｯｸにまとめ，ﾌﾞﾛｯｸ単位にｱｸｾｽする（ﾌﾞﾛｯｷﾝｸﾞ）
（ブロックに収まるレコード数にまとめる）
最後のｾｸﾀに余りが出るが，この領域は使用できない．この無駄を内部断片化と言う．

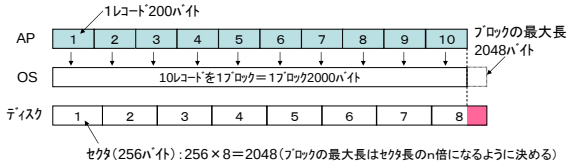


磁気ﾃﾞｨｽｸの記録量

1ｼﾘﾝﾀﾞあたりﾄﾗｯｸ数19，ﾄﾗｯｸあたりｾｸﾀ数42，1ｾｸﾀ256ﾊﾞｲﾄの磁気ﾃﾞｨｽｸに，1ﾚｺｰﾄﾞ200ﾊﾞｲﾄのﾚｺｰﾄﾞ10万件を順編成で格納したい．

①10レコードを1ブロックとして記録する時に必要なシリンダ数は？
　APは1ﾚｺｰﾄﾞずつOSに書き込み要求するが，OSは10ﾚｺｰﾄﾞまとめて書き込む
②1ブロック2000バイトでブロッキングする時に必要なシリンダ数は？
　APが要求した10レコード＝2000バイトのブロックをまとめて書き込む
③ブロック長（ブロックの最大長）が2048バイトの時に必要なシリンダ数は？
　OSはﾃﾞｨｽｸのｾｸﾀｻｲｽﾞの整数倍の単位でﾌﾞﾛｯｷﾝｸﾞを行う

①，②は同じ意味である．この問題では③も同じ意味になる．



磁気ディスクの記録量

以下の磁気ディスク装置に，長さ200バイトのレコード 10万件を順編成で格納したい．
シリンダあたりのトラック数19，トラックあたりのセクタ数42，1セクタ256B

パターン①
10レコードを 1ブロックとして記録するときに必要シリンダ数は幾つか．

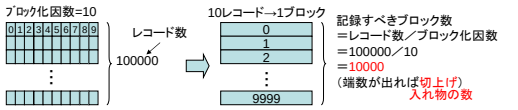
パターン②
ブロック長2048Bでブロッキングするときに必要なシリンダ数はいくつか．

第4回のスライド

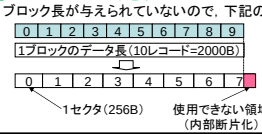
重要：磁気ディスクの記録量（1）

以下の磁気ディスク装置に，長さ200バイトのレコード 10万件を順編成で格納したい．
シリンダあたりのトラック数19，トラックあたりのセクタ数42，1セクタ256B
10レコードを 1ブロックとして記録するときに必要シリンダ数は幾つか．
基本情報（平成12年秋期）

①ブロック化因数（1ブロックに収容できるレコード数）：問題文で与えられている．
②記録すべきブロック数を求める（この値は⑤の計算で使うので後で計算しても良い）．



③1ブロックのセクタ数を求める
ブロック長が与えられていないので，下記の方法でセクタ数を求める．



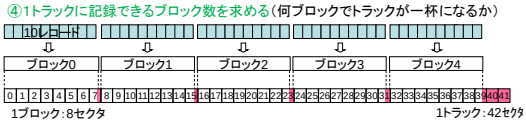
1ブロックのデータ長
=レコード長×ブロック化因数=200×10=2000
1ブロックのセクタ数＝データ長／セクタ長
＝2000/256=7.81→8（切上げ）
入れ物の数

参考　ブロック長 = 8セクタの長さ
= 256×8=2048　となる．

第4回のスライド

重要：磁気ディスクの記録量（2）

④1トラックに記録できるブロック数を求める（何ブロックでトラックが一杯になるか）



記録できるブロック数＝1トラックのセクタ数／1ブロックのセクタ数＝42／8＝5.25→5（切捨て）
トラックにブロックを入れるときのブロック数⇒入れるものの数

⑤最後に②，④の結果を使って，必要なトラック数を求め，シリンダ数を計算する

必要なトラック数=記録ブロック数／1トラックのブロック数=10000／5＝2000　（端数が出れば切上げ）
ブロック（複数）をトラックに入れるときのトラック数⇒入れ物の数

必要なシリンダ数=必要なトラック数／1シリンダのトラック数=2000／19＝105.26→106（切上げ）
トラック（複数）をシリンダに入れるときのシリンダ数⇒入れ物の数

切上げと切捨て
入れるものの数を求める：入れるものは入れ物よりも小さい→切捨て
入れ物の数を求める：入れ物は入れるものよりも大きい→切上げ

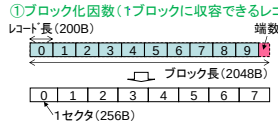
第4回のスライド

重要：磁気ディスクの記録量（3）

以下の磁気ディスク装置に，長さ200バイトのレコード 10万件を順編成で格納したい．
シリンダあたりのトラック数19，トラックあたりのセクタ数42，1セクタ256B
ブロック長2048Bでブロッキングするときに必要なシリンダ数はいくつか．

ブロック化因数の代わりにブロック長が与えられている

①ブロック化因数（1ブロックに収容できるレコード数）を求める



ブロック化因数
= ブロック長／レコード長
= 2048／200 = 10.24→10　（切捨て）
入れるものの数
（ブロックを複数のセクタに入れる時のセクタ数）

②以降は，前のスライドと同じ方法で求められる．
但し，③のセクタ数は，以下の計算で簡単に求まる．
セクタ数＝ブロック長／セクタ長＝2048/256=8

ディスクアクセスの最小単位は，セクタである．無駄が出ないように，
OSのアクセス単位であるブロックは，セクタ長の整数倍の長さに設定される．
ブロック化因数は，本スライドの①のように，ブロック長とレコード長により決まる
．
（あらかじめ，ブロック化因数が定められているわけではない）

（3）CPUｽｹｼﾞｭｰﾘﾝｸﾞｱﾙｺﾞﾘｽﾞﾑ

- 到着順ｻｰﾋﾞｽ：FCFS（First Come First Served）
  - ﾌﾟﾛｾｽの到着順に処理する
- 最短ｼﾞｮﾌﾞ優先：SJF（Shortest Job First）
  - CPUﾊﾞｰｽﾄ時間が短い順に処理する
- 優先式：（Priority Scheduling）
  - ﾌﾟﾛｾｽに予め優先度を与え、優先度順に処理
- ﾗｳﾝﾄﾞﾛﾋﾞﾝ：RR（Round Robin）
  - ﾌﾟﾛｾｽを順番に一定時間ずつ処理する
- 多重ﾚﾍﾞﾙ待ち列：（Multilevel Queue）
  - ﾌﾟﾛｾｽの特性に応じた複数の待ち列。待ち列間に優先度。
- 多重ﾚﾍﾞﾙﾌｨｰﾄﾞﾊﾞｯｸ列：（Multilevel Feedback Queue）
  - 時間内に終わらなければﾚﾍﾞﾙを落とす

- FCFSは、ﾉﾝﾌﾟﾘｴﾝﾌﾟﾃｨﾌﾞのみ。RR、多重ﾚﾍﾞﾙﾌｨｰﾄﾞﾊﾞｯｸ列は、ﾌﾟﾛｴﾝﾌﾟﾃｨﾌﾞのみ。他は、ﾌﾟﾘｴﾝﾌﾟﾃｨﾌﾞ、ﾉﾝﾌﾟﾘｴﾝﾌﾟﾃｨﾌﾞの2種類が可能。

CPUｽｹｼﾞｭｰﾘﾝｸﾞ（FCFS）

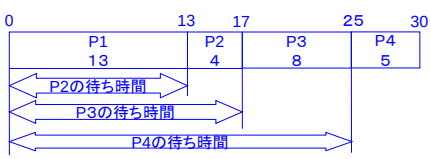
- 右表のプロセスが実行待ち列に並んでいる。時刻0でCPUが空きとなった。FCFSでスケジューリングする場合、時刻0からの平均待ち時間は？

ﾌﾟﾛｾｽ	ﾊﾞｰｽﾄ時間	到着順	
P1	13	1	
P2	4	2	
P3	8	3	
P4	5	4	

CPUｽｹｼﾞｭｰﾘﾝｸﾞ（FCFS）

- 右表のプロセスが実行待ち列に並んでいる。時刻0でCPUが空きとなった。FCFSでスケジューリングする場合、時刻0からの平均待ち時間は？

ﾌﾟﾛｾｽ	ﾊﾞｰｽﾄ時間	到着順	処理順
P1	13	1	1
P2	4	2	2
P3	8	3	3
P4	5	4	4



平均待ち時間
＝（0＋13＋17＋25）／4＝13．75
P1　P2　P3　P4　←ﾌﾟﾛｾｽ数

CPUｽｹｼﾞｭｰﾘﾝｸﾞ（SJF）

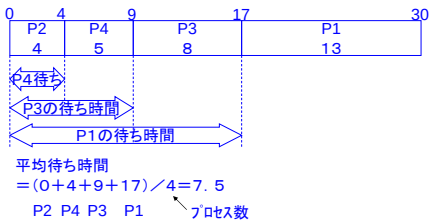
- 右表のプロセスが実行待ち列に並んでいる。時刻0でCPUが空きとなった。SJFでスケジューリングする場合、時刻0からの平均待ち時間は？

ﾌﾟﾛｾｽ	ﾊﾞｰｽﾄ時間	到着順	
P1	13	1	
P2	4	2	
P3	8	3	
P4	5	4	

CPUスケジューリング（SJF）

- 右表のプロセスが実行待ち列に並んでいる。時刻0でCPUが空きとなった。SJFでスケジューリングする場合、時刻0からの平均待ち時間は？

プロセス	バースト時間	到着順	処理順
P1	13	1	4
P2	4	2	1
P3	8	3	3
P4	5	4	2



平均待ち時間
＝（0＋4＋9＋17）／4＝7．5
P2 P4 P3 P1　←プロセス数

CPUスケジューリング（ラウンドロビン）

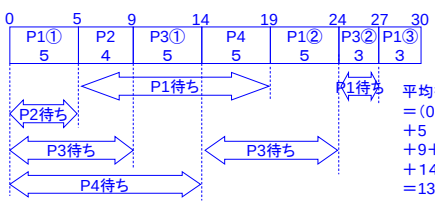
- 右表のプロセスが実行待ち列に並んでいる。時刻0でCPUが空きとなった。量子時間5のラウンドロビンでスケジューリングする場合、時刻0からの平均待ち時間は？

プロセス	バースト時間	到着順	
P1	13	1	
P2	4	2	
P3	8	3	
P4	5	4	

CPUスケジューリング（ラウンドロビン）

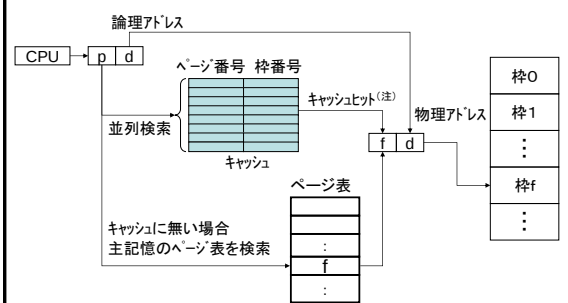
到着順に処理を始めるが、量子時間で終わらなければ次プロセスと交代する。交代してから、再度処理するまでの時間も待ち時間になる。

プロセス	バースト時間	到着順	処理順
P1	13	1	1,5,7
P2	4	2	2
P3	8	3	3,6
P4	5	4	4



平均待ち時間
＝（0＋14＋3＋5＋9＋10＋14）／4
＝13.75
P1
P2
P3
P4

（4）キャッシュを用いたページングハードウェア



キャッシュヒット時：キャッシュアクセス時間＋主記憶（物理記憶）アクセス時間
ミスヒット時：キャッシュアクセス時間＋主記憶（ページ表）アクセス時間＋主記憶（物理記憶）アクセス時間

キャッシュを用いたアクセス時間

- ページングにおいて、キャッシュヒット時のアクセス時間を10ナノ秒、ヒットしない時のアクセス時間を22ナノ秒とする。ヒット率を80%とした場合の平均アクセス時間は、何ナノ秒か。

キャッシュを用いたアクセス時間

- ページングにおいて、キャッシュヒット時のアクセス時間を10ナノ秒、ヒットしない時のアクセス時間を22ナノ秒とする。ヒット率を80%とした場合の平均アクセス時間は、何ナノ秒か。

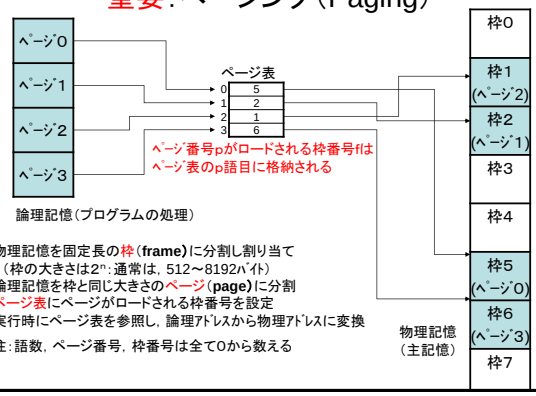
EAT＝Ap＋B（1－p）　p：キャッシュヒット率
A：ヒット時のアクセス時間
B：ヒットしない時のアクセス時間
EAT＝10＊0．8＋22＊（1－0．8）＝8＋4．4＝12．4

(5)ページングと仮想記憶

- 文章中の穴埋め：選択式（基本的な用語の理解）
  - 論理記憶＝ページ（中身）　論理アドレス＝ページ番号・変位
  - 物理記憶＝枠（入れ物）　物理アドレス＝枠番号・変位
  - ページング方式、仮想記憶の構成と動作
    - ページフォールト，割込みの種類
    - ページアウト、ページイン
    - プロセスの状態遷移、切り替え
    - 外部断片化，動的再配置
- アドレス計算
  - 論理アドレス5個の物理アドレス
  - 内，枠が割り当てられていない論理アドレスを識別
  - 2進数／10進数

p.128

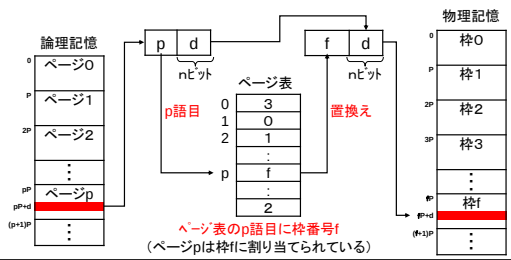
重要：ページング（Paging）



物理記憶を固定長の枠（**frame**）に分割し割り当て
（枠の大きさは2^n：通常は，512～8192ﾊﾞｲﾄ）
論理記憶を枠と同じ大きさのページ（**page**）に分割
ページ表にページがロードされる枠番号を設定
実行時にページ表を参照し，論理ｱﾄﾞﾚｽから物理ｱﾄﾞﾚｽに変換
注：語数，ページ番号，枠番号は全て0から数える

重要：ページングにおけるアドレス変換

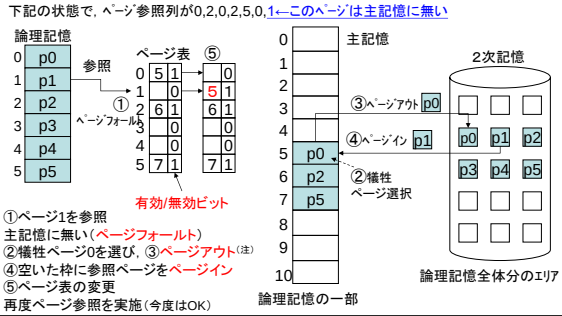
論理アドレスAを物理アドレスBに変換（ページサイズP＝2^nとする）
①論理アドレスAのページ番号pと変位d（ページの先頭から何語目か）を求める
　p=A div P（AをPで割った商），d= A mod P（AをPで割った余り）
　（2進数で表すとpはn+1ﾋﾞｯﾄ目から上位，dは下位nﾋﾞｯﾄになる）
②論理アドレスA＝pP+dの「p」をページ表のp語目の値「f」（枠番号）で置き換える．
③物理アドレスB＝fP＋d（2進数で表すと，pのビットをfのビットに置き換えただけになる）



仮想記憶の概念

論理記憶の方が割り当てられた物理記憶（主記憶）よりも大きい
・2次記憶上に論理記憶の全体（全てのページ）が収容できるエリアを確保
・主記憶上に論理記憶の一部（当面の計算に必要な分だけ）を置く

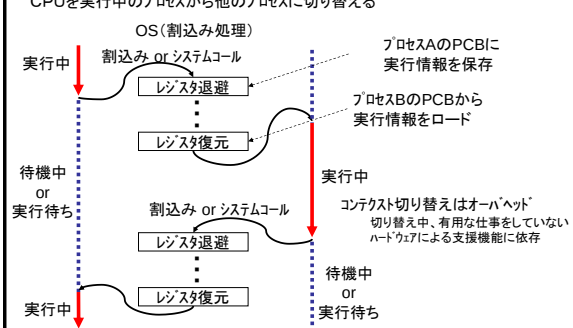
下記の状態で，ﾍﾟｰｼﾞ参照列が0,2,0,2,5,0,1←このﾍﾟｰｼﾞは主記憶に無い



①ページ1を参照
主記憶に無い（ページフォールト）
②犠牲ページ0を選び，③ページアウト(注)
④空いた枠に参照ページをページイン
⑤ページ表の変更
再度ページ参照を実施（今度はOK）

ｺﾝﾃｸｽﾄ切替え

CPUを実行中のﾌﾟﾛｾｽから他のﾌﾟﾛｾｽに切り替える



仮想記憶の実効ｱｸｾｽ時間

主記憶のアクセス時間を2ﾏｲｸﾛ秒，ﾍﾟｰｼﾞｲﾝ・ﾍﾟｰｼﾞｱｳﾄ時間の平均を10ﾐﾘ秒とする．ﾍﾟｰｼﾞﾌｫｰﾙﾄの確率が5×10^{-6}の場合の実効アクセス時間を求めよ．

仮想記憶の実効ｱｸｾｽ時間

主記憶のアクセス時間を2ﾏｲｸﾛ秒，ﾍﾟｰｼﾞｲﾝ・ﾍﾟｰｼﾞｱｳﾄ時間の平均を10ﾐﾘ秒とする．ﾍﾟｰｼﾞﾌｫｰﾙﾄの確率が5×10^{-6}の場合の実効アクセス時間を求めよ．

EAT＝$10\times10^{3}\times5\times10^{-6}+2\times(1-5\times10^{-6})$
＝0．05＋2－0．00001＝2．04999［ﾏｲｸﾛ秒］

実効ｱｸｾｽ時間　EAT=Ap＋B（1－p）
A:ﾍﾟｰｼﾞﾌｫｰﾙﾄ時の処理時間（主にﾍﾟｰｼﾞｱｳﾄ，ﾍﾟｰｼﾞｲﾝ時間）
B:ﾍﾟｰｼﾞﾌｫｰﾙﾄしない時の処理時間（主記憶ｱｸｾｽ時間）

アドレス変換（10進数）

1ページ16語の仮想記憶システムがあり、ページ表は表のようになっている。論理アドレス50番地に対応する物理アドレスの番地を求めよ．

7	1
41	1
	0
14	1

有効／無効ビット

アドレス変換（10進数）

1ページ16語の仮想記憶システムがあり、ページ表は表[添付ファイル表1]のようになっている。論理アドレス50番地に対応する物理アドレスの番地を求めよ．[数値のみを半角数字で記入]

答　226

論理ｱﾄﾞﾚｽA=50
ﾍﾟｰｼﾞｻｲｽﾞP=16
p=A div P=3
d=A mod P =2
(50÷16=3･･･2)

ﾍﾟｰｼﾞ番号p　dはそのまま

3	2

変位d

14	2

枠番号f　変位d

物理ｱﾄﾞﾚｽB
=f×P+d
=14×16+2
=226

p=3

0	7	1
1	41	1
2		0
3	14	1

f=14に置き換え

有効／無効ビット

アドレス変換とページフォールト

1ページ16語の仮想記憶システムがあり、ページ表は下のようになっている。表1の論理アドレスに対応する物理アドレスを求めよ。（対応する物理ｱﾄﾞﾚｽが存在しない場合は「－」を記入）．

ページ表

7	1
41	1
	0
14	1

有効／無効ビット

表1

論理ｱﾄﾞﾚｽ
6
15
30
40
50

アドレス変換とページフォールト

1ページ16語の仮想記憶システムがあり、ページ表は下のようになっている。表1の論理アドレスに対応する物理アドレスを求めよ。（対応する物理ｱﾄﾞﾚｽが存在しない場合は「－」を記入）．

ページ表

0	7	1
1	41	1
2		0
3	14	1

有効／無効ビット

表1

論理ｱﾄﾞﾚｽ	ﾍﾟｰｼﾞ番号 ↓	枠番号 ↓	物理ｱﾄﾞﾚｽ計算 B=f×P+d	物理ｱﾄﾞﾚｽ
6	6÷16=0･･･6	7･･･6	7×16＋ 6=	118
15	15÷16=0･･･15	7･･･15	7×16+15=	127
30	30÷16=1･･･14	41･･･14	41×16+14=	670
40	40÷16=2･･･8	存在しない	ページフォールト	－
50	50÷16=3･･･2	14･･･2	14×16＋ 2=	226

ﾍﾟｰｼﾞ番号→枠番号

アドレス変換（2進数）

1ページ16語の仮想記憶システムがあり、ページ表は下のようになっている。論理アドレス110010番地に対応する物理アドレスの番地を求めよ．尚，アドレスとページ表の値は2進数である．

ページ表

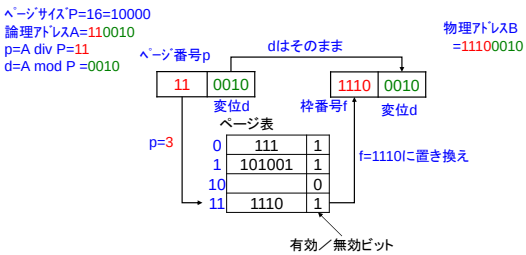
111	1
101001	1
	0
1110	1

有効／無効ビット

アドレス変換（2進数）

1ページ16語の仮想記憶システムがあり、ページ表は下のようになっている。論理アドレス110010番地に対応する物理アドレスの番地を求めよ．尚，アドレスとページ表の値は2進数である．

答　11100010



アドレス変換（2進数）

1ページ16語の仮想記憶システムがあり、ページ表は表のようになっている。論理アドレス101番地に対応する物理アドレスの番地を求めよ．尚，アドレスとページ表の値は2進数である．

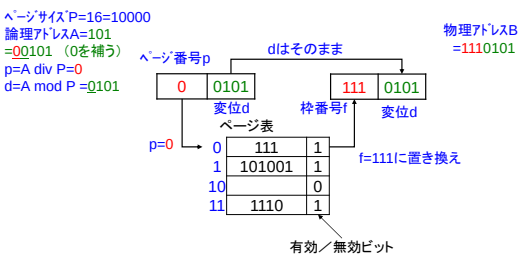
111	1
101001	1
	0
1110	1

有効／無効ビット

アドレス変換（2進数）

1ページ16語の仮想記憶システムがあり、ページ表は表のようになっている。論理アドレス101番地に対応する物理アドレスの番地を求めよ．尚，アドレスとページ表の値は2進数である．

答　1110101



アドレス変換とページフォールト

1ページ16語の仮想記憶システムがあり、ページ表は下のようになっている。表1の論理アドレスに対応する物理アドレスを求めよ。（対応する物理ｱﾄﾞﾚｽが存在しない場合は「ー」を記入）．尚，アドレスとページ表の値は2進数である．

111	1
101001	1
	0
1110	1

有効／無効ビット

論理ｱﾄﾞﾚｽ
110
1111
11110
101000
110010

アドレス変換とページフォールト

1ページ16語の仮想記憶システムがあり、ページ表は下のようになっている。表1の論理アドレスに対応する物理アドレスを求めよ。（対応する物理ｱﾄﾞﾚｽが存在しない場合は「ー」を記入）．尚，アドレスとページ表の値は2進数である．

0	111	1
1	101001	1
10		0
11	1110	1

有効／無効ビット

論理ｱﾄﾞﾚｽ	ﾍﾟｰｼﾞ番号	変位	枠番号	変位	物理ｱﾄﾞﾚｽ 枠番号と変位を結合
110	0	0110	111	0110	1110110
1111	0	1111	111	1111	1111111
11110	1	1110	101001	1110	1010011110
101000	10	1000	―		―
110010	11	0010	1110	0010	11100010

試験までに準備すべき事項

先ず，本日配布したプリントの問題の解法を理解すること．
以下の小テストを必ず復習すること（試験との関連が大きい）．
（設問のみを見て，自分の手で答えが書けるようにしておくこと）
文章問題（プリントには無い）は，解説もきちんと読んでおくこと．

ファイルシステム（シリンダ数の計算）
基礎OS2015④問1～5，⑤問1～4

プロセスの状態遷移，CPUスケジューリング
基礎OS2015⑨問1～10
基礎OS2015⑩問1～7

仮想記憶
基礎OS2015⑪問2，3
基礎OS2015⑫問1～10
基礎OS2015⑬問1～7