

## 基礎OS⑩ CPUスケジューリング， メモリの管理(1)

2012年度(3時限目)

### 問1～4

図1

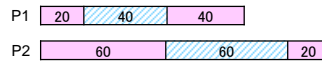


表1

時刻	発生した事象	P1	P2
0	CPU割当て(P1)		
A	(1)I/O要求 (2)CPU割当て		
B	(3)I/O完了		
C	(4)I/O要求 (5)CPU割当て		
D	(6)終了		
E	(7)I/O完了 (8)CPU割当て		
F	(9)終了		

### 問5～7

表2

プロセス	バースト時間	到着順
P1	13	1
P2	3	2
P3	15	3
P4	7	4

### 問9

表3 ロードモジュールの大きさ

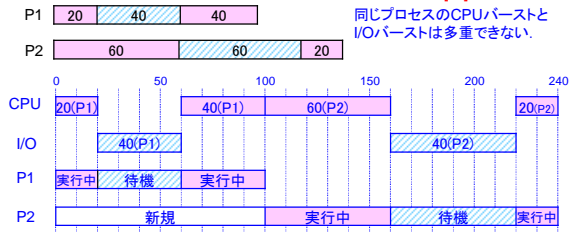
プロセス	大きさ(kバイト)
A	200
B	100
C	150
D	80
E	90

## 問1 プロセスの実行(1)

単独で実行するとスライド(図1)のような動作をするプロセスP1, P2がある。  
これら2つを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。

答 58 [%]

同じプロセスのCPUバーストとI/Oバーストは多重できない。



実行時間: 20+40+40+60+60+20=240  
CPU使用時間: 20+40+60+20=140 (CPUバースト時間の合計)  
CPU使用率: 140/240=0.5833≒58%

## 問1 プロセスの実行と発生する事象

**I/O要求:** プログラム中の入出力要求命令を実行した時に発生する。

**終了:** プログラム中の最後の命令を実行した時に発生する。

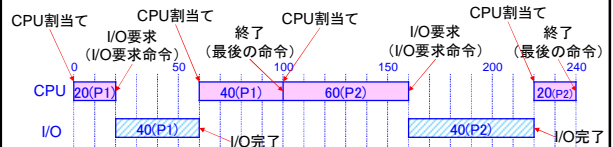
(従って、これらの事象は実行中状態のプロセスに対してのみ発生する)

**I/O完了:** 入出力装置の動作が終了した時に発生する

(待機状態(=I/Oバースト中)のプロセスに対してのみ発生する)

**CPU割当て:** CPUが空いていて、レディ状態のプロセスがある場合に発生する。

(レディ状態のプロセスに対してのみ発生する)



多重動作する場合でも、プロセス単体では上記と同じ事象が同じ順序で発生している。  
即ち、多重動作を理解するためには、プロセス単体での発生事象と状態遷移を理解しておく必要がある。

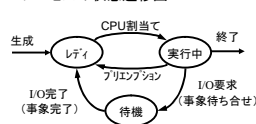
## 問2 プロセスの実行(2)

スライド(図1)のプロセスを多重して動作させる。プロセスP1, P2ともに生成が終了しており、実行待ち列に並んでいる。時刻0で、P1にCPUが割り当てられた後、表1(図1)のような事象が発生した。A～Fの時刻を記入せよ。[A～Fの欄に半角の「」で時刻を、半角数字で記入。「」は記入しないこと。空白欄と余白な文字を記入しないこと。例: 10,20,30,40,50,60]

答 20,60,80,120,140,160

時刻	発生した事象	P1	P2
0	CPU割当て(P1)		
A	(1)I/O要求 (2)CPU割当て		
B	(3)I/O完了		
C	(4)I/O要求 (5)CPU割当て		
D	(6)終了		
E	(7)I/O完了 (8)CPU割当て		
F	(9)終了		

プロセスの状態遷移図



状態と発生し得る事象から、どちらのプロセスに発生した事象かを判定する

## 問3 プロセスの実行(3)

問2において、(4)(5)の事象が発生した後のP1, P2の状態は？ [P1, P2の欄に状態名のみを全角の「」で取得して全角文字で記入(I/O状態の「待機」は記入してはならない。「」は記入してはならない)。今回の「」は全角の「」で注意。]

時刻	発生した事象	P1	P2
0	CPU割当て(P1)		
A	(1)I/O要求 (2)CPU割当て		
B	(3)I/O完了		
C	(4)I/O要求 (5)CPU割当て		
D	(6)終了		
E	(7)I/O完了 (8)CPU割当て		
F	(9)終了		

時刻0より順に、状態遷移を記入していくこと。  
入出力要求は、実行中のプロセスにおける事象。  
実行中(入出力要求)→待機  
入出力完了は待機状態のプロセスにおける事象。  
待機(入出力完了)→レディ  
CPU割当てはレディ状態のプロセスにおける事象。  
レディ(CPU割当て)→実行中

答 実行中, 待機

## 問1～4 発生した事象と状態遷移

時刻	発生した事象	P1	P2
0	CPU割当て(P1)	実行中	レディ
20	(1)I/O要求 (2)CPU割当て	待機 待機	レディ 実行中
60	(3)I/O完了	レディ	実行中
80	(4)I/O要求 (5)CPU割当て	レディ 実行中	待機 待機
120	(6)終了 (停止)	(停止)	待機
140	(7)I/O完了 (8)CPU割当て		レディ 実行中
160	(9)終了		(停止)

## 問4 プロセスの実行(4)

問2において、時刻0から両方が終了するまでの間のCPU使用率[%]を求めよ。【数値のみを半角数字の整数値で記入(計算結果が小数の場合は、小数点以下を四捨五入)。】

答 88 [%]

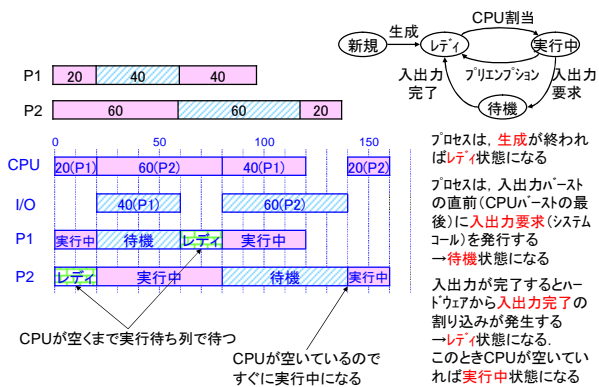
CPU使用率=140/160=0.875(87.5%)

時刻	発生した事象	P1	P2
0	CPU割当て(P1)	実行中	レディ
20	(1)I/O要求 (2)CPU割当て	待機 待機	レディ 実行中
60	(3)I/O完了	レディ	実行中
80	(4)I/O要求 (5)CPU割当て	レディ 実行中	待機 待機
120	(6)終了 (停止)	(停止)	待機
140	(7)I/O完了 (8)CPU割当て		レディ 実行中
160	(9)終了		(停止)

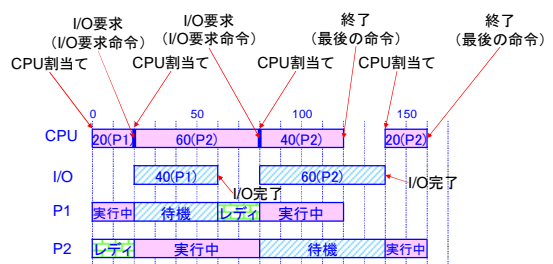
開始から終了までの時間=160  
時刻120～140の間、CPU不使用  
(CPU使用時間=160-20=140)

プロセスの実行については、次スライドを参照

## プロセスの実行と状態



## 問2～4 プロセスの実行と発生した事象



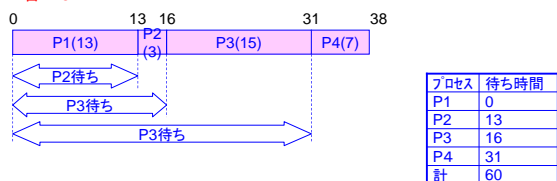
## 問5 FCFSの平均待ち時間

スライド【問1の添付ファイル】表2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、平均実行待ち時間を求めよ。

【数値のみを半角数字で記入(小数点も半角)。値が整数の場合は整数値で記入。】  
先着順(P1, P2, P3, P4の順)に処理される  
下図より待ち時間の合計は60

平均待ち時間=60/4=15  
(待ち時間の合計をプロセス数で割る)

答 15



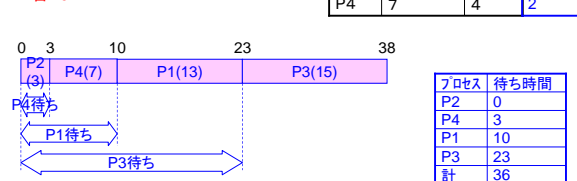
## 問6 SJFの平均待ち時間

スライド【問1の添付ファイル】表2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、平均実行待ち時間を求めよ。

【数値のみを半角数字で記入(小数点も半角)。値が整数の場合は整数値で記入。】  
先着順(P1, P2, P3, P4の順)に処理される  
下図より待ち時間の合計は36

平均待ち時間=36/4=9  
(待ち時間の合計をプロセス数で割る)

答 9

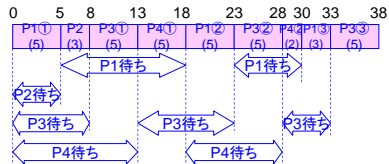


## 問7 ラウンドロビンの平均待ち時間

スライド【問1の添付ファイル】表2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、平均実行待ち時間を求めよ。【括弧のみを半角数字で記入(小数点も半角)。値が整数の場合は整数値で記入。】

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、再び処理されるまでの時間も待ち時間になる。下図より待ち時間の合計は71  
平均待ち時間=71/4=17.75  
(待ち時間の合計をプロセス数で割る)

答 17.75



プロセス	バースト時間	到着順	処理順
P1	13	1	1,5,8
P2	3	2	2
P3	15	3	3,6,9
P4	7	4	4,7

プロセス	待ち時間
P1	20(0+13+7)
P2	5
P3	23(8+10+5)
P4	23(13+10)
計	71

## 問8 再配置

プログラムを実行する場合、物理記憶へのロード位置に対応してプログラム内のアドレス情報の補正が必要である。これを表す用語は、以下のどれか。(第2種 平成8年度・秋期, 基本情報平成19年度・秋期)

- A. コンパイル
- B. 再配置**
- C. 最適化
- D. リロード
- E. リンケージ

再配置: プログラムを実行するために、ロード位置に対応してアドレス情報を補正  
静的再配置: 主記憶へのロード時に再配置ローダがアドレスの補正をする  
(物理アドレスに書き換える)

動的再配置: ロードモジュールを論理アドレスで作成。実行時に物理アドレスに変換。  
(主記憶には、論理アドレスのままロードされる)

## 問9 記憶管理(外部断片化)

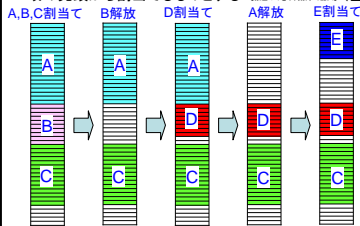
主記憶を可変区画割当てするシステムがある。プロセスA~Eはスライド【問1の添付ファイル】表3の大きさのモジュールを使用する。OSは、次の順序で割当て、解放する。

A割当て→B割当て→C割当て→B解放→D割当て→A解放→E割当て

最後のE割当ての時点で、主記憶の空き領域は何か所存在するか。

尚、主記憶は500kバイトで初期状態では全て空いている。また、モジュールは空き領域の先頭から割当てするものとする。【括弧のみを半角数字で記入】(基本情報 平成19年度・春期)

答 3



プロセス	大きさ(kバイト)
A	200
B	100
C	150
D	80
E	90

穴の大きさ[kバイト]: 110, 20, 50  
空き領域合計: 180

外部断片化: 主記憶を可変区画割当てする場合、割り当て、開放の繰り返しにより、小さな穴が多数できる問題

## 問10 再配置レジスタ

プログラムが実行中に参照した論理アドレス200番地は、主記憶の9392番地にロードされている。この時の再配置レジスタの値は幾つか。

答 9192

主記憶上の番地(物理アドレス)=論理アドレス+再配置レジスタの内容  
再配置レジスタの内容=9392-200=9192

