

# オペレーティングシステム

## 第4章 スケジューリング

<https://github.com/tctsigemura/OSTextBook>

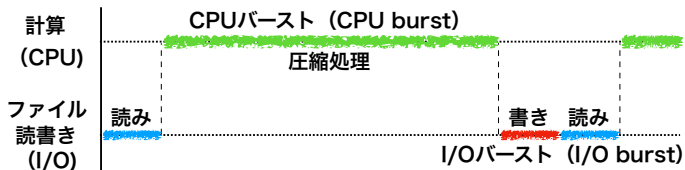
- スループット (Throughput)
- ターンアラウンド時間 (Turnaround time)
- レスポンス時間 (Response time)
- 締め切り (Deadline)
- その他 (公平性, 省エネ, 予測性など)

# システムごとの目標

コンピュータの種類	重視する性能
メインフレーム (バッチ処理)	スループット, ターンアラウンド時間
ネットワークサーバ	レスポンス時間, スループット
デスクトップパソコン	レスポンス時間
モバイルデバイス	レスポンス時間, 省エネルギー
組込み制御	締め切り

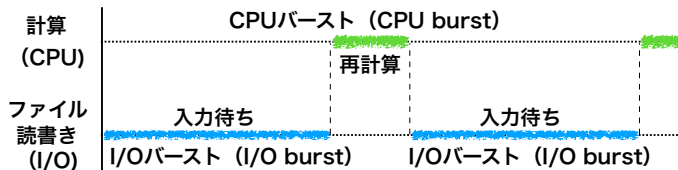
# CPU バウンドプロセス

## 動画圧縮の例



# I/O バウンドプロセス

## スプレッドシートの例

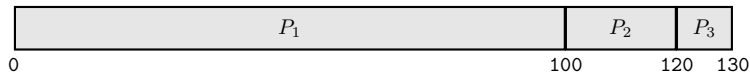


# FCFS スケジューリング (1)

## FCFS (First-Come, First-Served)

- プリエンプションしないスケジューリング方式

プロセス	到着時刻	CPU バースト時間 (ms)
$P_1$	0	100
$P_2$	0	20
$P_3$	0	10

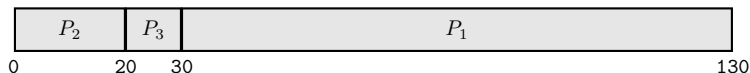


- $P_1$ ,  $P_2$ ,  $P_3$  の順に実行
- 平均ターンアラウンド時間  $((100 + 120 + 130)/3 = 117 \text{ ms})$
- 最悪の平均ターンアラウンド時間を選択することもある。

## FCFS スケジューリング (2)

- 平均ターンアラウンド時間は 到着順により大きく変化する.

プロセス	到着時刻	CPU バースト時間 (ms)
$P_1$	0	100
$P_2$	0	20
$P_3$	0	10



- $P_2$ ,  $P_3$ ,  $P_1$  の順に実行
- 平均ターンアラウンド時間  $((20 + 30 + 130)/3 = 60 \text{ ms})$

# SJF スケジューリング

## SJF (Shortest-Job-First)

- プリエンプションしないスケジューリング方式

プロセス	到着時刻	CPU バースト時間 (ms)
$P_1$	0	100
$P_2$	0	20
$P_3$	0	10



- 平均ターンアラウンド時間  $((10 + 30 + 130)/3 = 57 \text{ ms})$

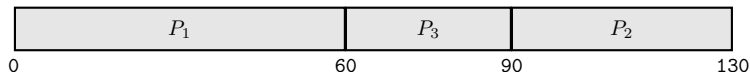


# SRTF スケジューリング (1)

SRTF (Shortest-Remaining-Time-First)

比較のための SJF スケジューリングの例

プロセス	到着時刻	CPU バースト時間 (ms)
$P_1$	0	60
$P_2$	10	40
$P_3$	60	30



- SJF はプリエンプションなし
- 平均ターンアラウンド時間  
$$(((60 - 0) + (90 - 10) + (130 - 60))/3 = 70 \text{ ms})$$

# SRTF スケジューリング (2)

SRTF (Shortest-Remaining-Time-First)

前の SJF と同じプロセスのを SRTF でスケジューリング

プロセス	到着時刻	CPU バースト時間 (ms)
$P_1$	0	60
$P_2$	10	40
$P_3$	60	30



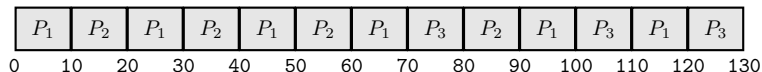
- SRTF はプリエンプションあり
- 平均ターンアラウンド時間  
$$(((130 - 0) + (50 - 10) + (90 - 60))/3 = 67 \text{ ms})$$

# RR スケジューリング (1)

## RR (Round-Robin)

クオンタムタイムまでプリアンプションしない。

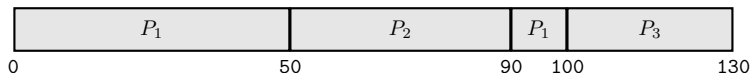
プロセス	到着時刻	CPU バースト時間 (ms)
$P_1$	0	60
$P_2$	10	40
$P_3$	60	30



- クオンタムタイム = 10ms
- 平均ターンアラウンド時間  
 $((120 - 0) + (90 - 10) + (130 - 60)) / 3 = 90$

## RR スケジューリング (2)

プロセス	到着時刻	CPU バースト時間 (ms)
$P_1$	0	60
$P_2$	10	40
$P_3$	60	30

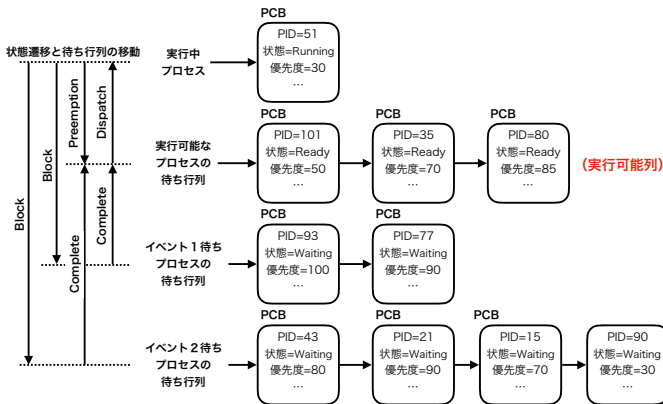


- クォンタムタイム = 50ms
- 平均ターンアラウンド時間  
 $((100 - 0) + (90 - 10) + (130 - 60)) / 3 = 83 \text{ ms}$

# 優先度順スケジューリング

## Priority

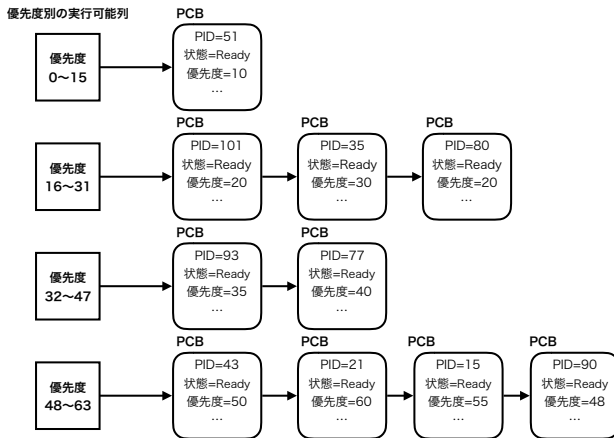
- 実行可能列を優先度順でソートしておく。



- 静的優先度／動的優先度
- スタベーション (starvation)：飢餓
- エージング (aging)：老化，熟成

# FB スケジューリング

## FB (Multilevel Feedback Queue)



### ● エージング

# 第19章（6節） TacOSのスケジューラ

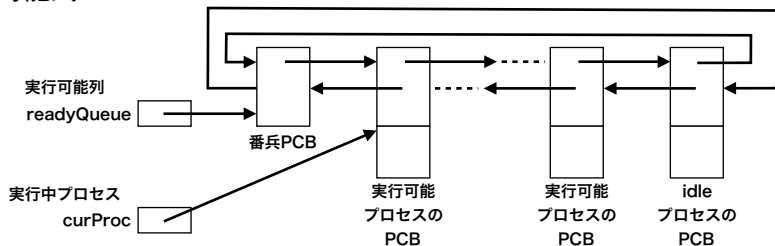
# TacOS のスケジューラ

```
1 public void schProc(PCB proc) {  
2     int r = setPri(DI|KERN);           // 割り込み禁止、カーネル  
3     int enice = proc.enice;  
4     PCB head = readyQueue.next;        // 実行可能列から  
5     while (head.enice<=enice)           // 優先度がより低い  
6         head = head.next;              // プロセスを探す  
7     insProc(head,proc);                 // 見つけたプロセスの  
8     setPri(r);                          // 直前に挿入する  
9 }
```



# TacOS の実行可能列 (参考)

- yield()
- dispatch()
- 実行可能列



## 練習問題

# 練習問題（１）

- 次の言葉の意味を説明しなさい。
  - スループット
  - ターンアラウンド時間・レスポンス時間
  - ハードリアルタイム・ソフトリアルタイム
  - CPU バウンドプロセス・I/O バウンドプロセス
  - FCFS スケジューリング・SJF スケジューリング
  - SRTF スケジューリング・RR スケジューリング
  - 優先度順スケジューリング・FB スケジューリング
  - クォンタム時間
  - スタベーション
  - エージング

## 練習問題 (2)

- 次の三つのプロセスの実行順をガントチャートで示しなさい。また、平均ターンアラウンド時間を計算しなさい。

プロセス名	到着時刻 (ms)	CPU バースト時間 (ms)
$P_1$	0	70
$P_2$	10	50
$P_3$	20	30

- FCFS でスケジューリングした場合
- SJF でスケジューリングした場合
- SRTF でスケジューリングした場合
- RR (但しクォンタム時間は 20ms) でスケジューリングした場合
- RR (但しクォンタム時間は 40ms) でスケジューリングした場合
- RR (但しクォンタム時間は 60ms) でスケジューリングした場合