

# オペレーティングシステム

## 第4章 スケジューリング

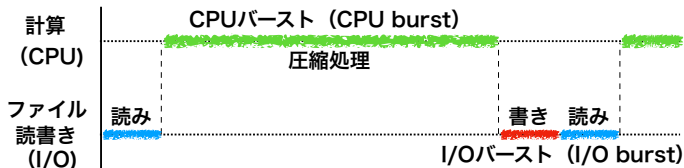
<https://github.com/tctsigemura/OSTextBook>

- 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)

# スケジューリングの目標

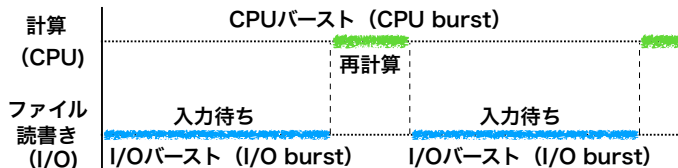
コンピュータの種類	重視する性能
メインフレーム (バッチ処理)	スループット, ターンアラウンド時間
ネットワークサーバ	レスポンス時間, スループット
デスクトップパソコン	レスポンス時間
モバイルデバイス	レスポンス時間, 省エネルギー
組込み制御	締め切り

# CPU バウンドプロセス



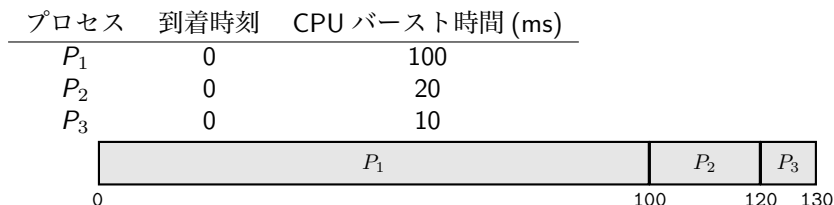
- 動画圧縮の例
- I/O バウンドプロセス (エクセル)

# I/O バウンドプロセス



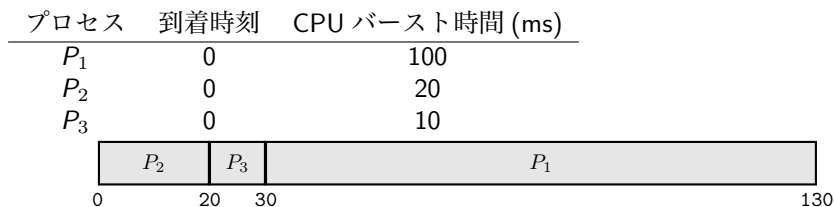
- スプレッドシートの例

# FCFS スケジューリング (1)



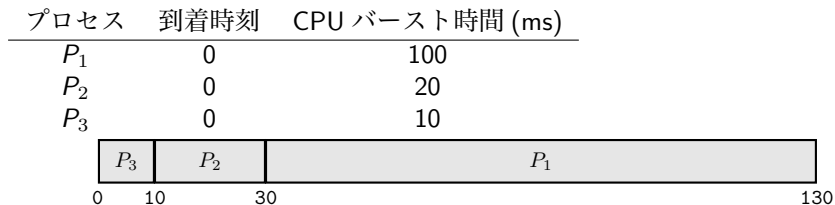
- $P_1, P_2, P_3$  の順に実行
- 平均ターンアラウンド時間  $((100 + 120 + 130)/3 = 117 \text{ ms})$

# FCFS スケジューリング (2)



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

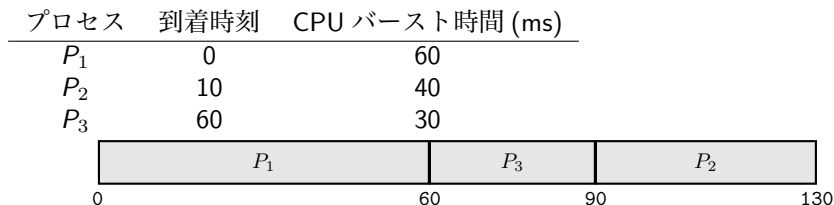
# SJF スケジューリング



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

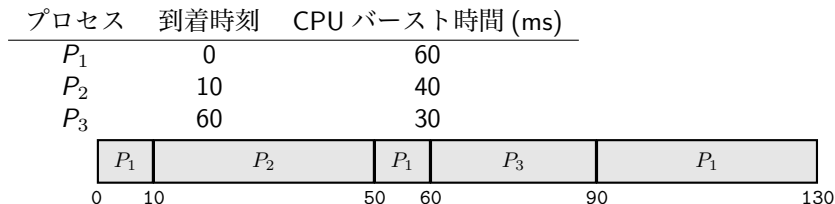


# SJF スケジューリング (比較のため)



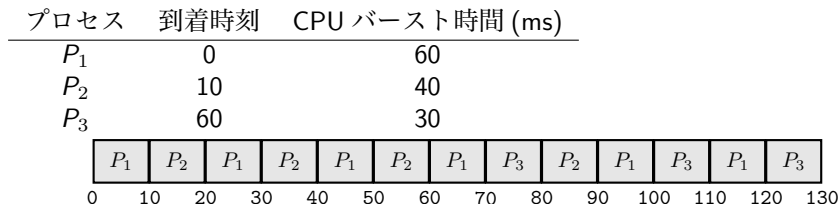
- SJF はプリエンプションなし
- 平均ターンアラウンド時間  
 $((60 - 0) + (90 - 10) + (130 - 60))/3 = 70 \text{ ms}$

# SRTF スケジューリング



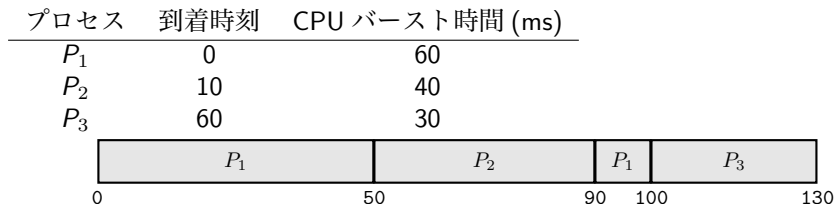
- SRTF はプリエンプションあり
- 平均ターンアラウンド時間  
 $((130 - 0) + (50 - 10) + (90 - 60))/3 = 67 \text{ ms}$

# RR スケジューリング (1)



- クォンタムタイム = 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)

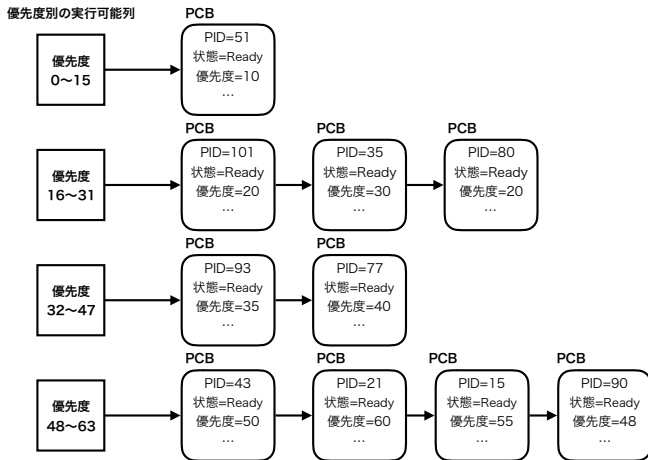


- クォンタムタイム = 50ms
- 平均ターンアラウンド時間  
 $((100 - 0) + (90 - 10) + (130 - 60)) / 3 = 83 \text{ ms}$

# 優先度順スケジューリング

- 静的・動的
- スタベーション
- エージング

# FB スケジューリング



## ● エーシング

# 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

