

プロセス



佐藤一郎

2001年5月29日

Ichiro Satoh

オペレーティングシステムの構成

オペレーティングシステムの基本部分をカーネルと呼ぶ



システムプログラム:

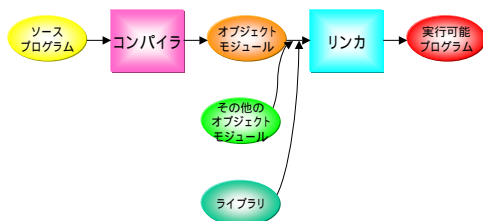
- コンパイラ、リンカ、エディタなどはユーザプログラムとして実行される。
- ソフトウェア階層としてはユーザプログラムとカーネルの中間

カーネル

Ichiro Satoh

プログラムのコンパイル

ユーザプログラムは複数ステップを経て実行可能プログラムとなる



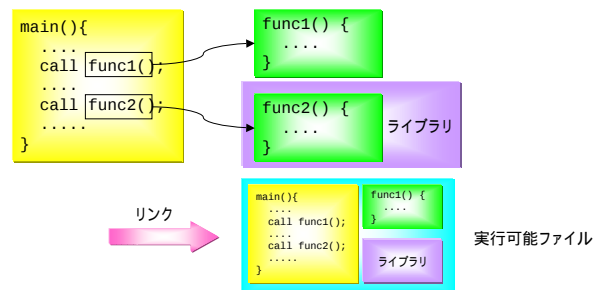
コンパイラ: C言語やPascal言語などの高水準言語を機械語(オブジェクト)に変換する
リンカ: 機械語によるオブジェクトモジュールを合成して実行可能なプログラムを生成する

Ichiro Satoh

静的リンクング

プログラム開発では複数モジュールから構成

→ ロード・実行する前に必要なモジュールを統合する必要



Ichiro Satoh

▶ ローディング

オペレーティングシステムはメモリ上のプログラムを所定のアドレスから命令・データを読み込み実行する。

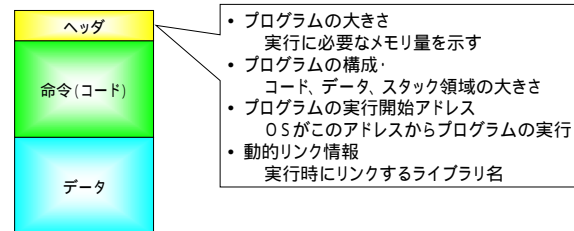
ローディング手順:

1. プログラムを格納しているファイルを見つける
2. プログラムを実行するプロセスを生成する
3. プログラムを実行に必要なメモリ領域を確保する
4. ファイル内のプログラム(命令とデータ)を読み込む
5. 命令とデータを特定のメモリアドレスに配置
6. パラメータを渡し、所定のアドレスから実行を開始する

Ichiro Satoh

▶ プログラムの構造

実行可能プログラム(ファイル)の構成

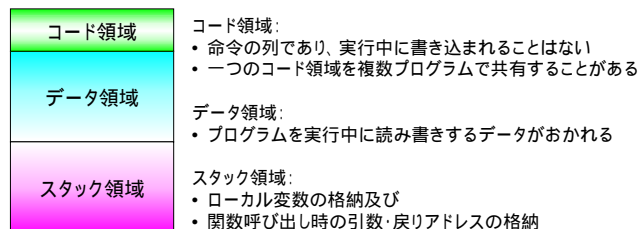


UNIXでは通称 a.out などと呼ばれるファイル

Ichiro Satoh

▶ メモリ領域

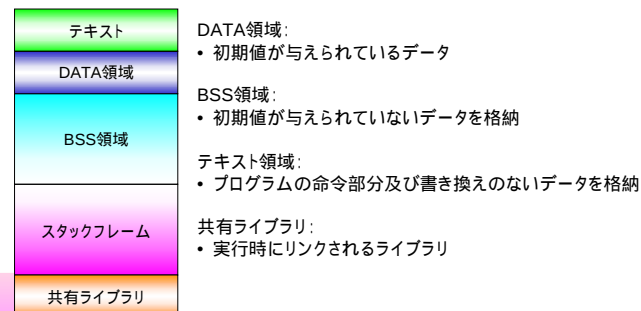
プログラムの実行領域は、コード、データ、スタックの3領域から構成



Ichiro Satoh

▶ メモリ領域(UNIXの場合)

データ領域はDATA領域とBSS領域から構成



Ichiro Satoh

▶ プロセス

コンピュータ上でプログラムを実行するための実体

c.f. タスクやジョブと呼ぶこともある

例: ワープロやWWWブラウザの実行ではそれぞれがプロセス内で実行

プロセスは仮想的なCPUを通じて実行



Ichiro Satoh

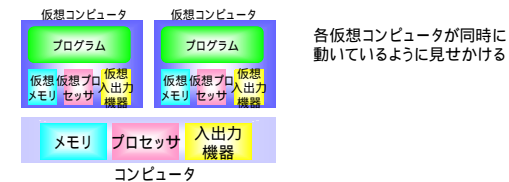
▶ マルチタスク

演算処理装置は一度に一つのプログラムを実行(プロセス)

問題: 実行中の処理が終了しないと次の処理が行えない

擬似的に複数のプログラムを同時に実行しているように見せかける

→ マルチタスキング



Ichiro Satoh

▶ マルチタスク処理の実現方法

並行処理:

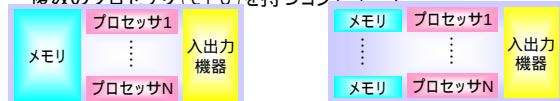
複数のプロセスを同時に動作させること(と見せかけること)

並列処理:

複数のプロセッサ(CPU)により複数のプロセスを同時に動作させること

c.f. 並列コンピュータ

複数のプロセッサ(CPU)を持つコンピュータ



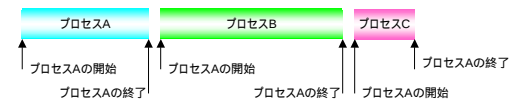
共有メモリ型並列コンピュータ

分散メモリ型並列コンピュータ

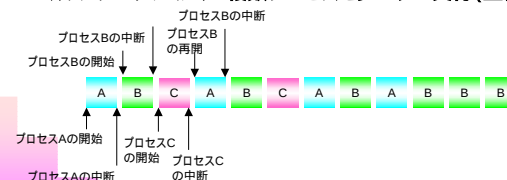
Ichiro Satoh

▶ 並行処理

逐次実行: 一つずつ順番に実行



マルチプログラミング: 複数プロセスを少しずつ実行(並行実行)



Ichiro Satoh

プロセスの切り替え

OSは複数のプロセスを並行して実行するため、プロセスを切り替えてプロセッサ(CPU)を割り当てる

プロセスの切り替え:

- あるプロセスの実行を一時的に中断し、次に実行すべきプロセスを選んで実行させること
- c.f. コンテキスト切り替え

ディスパッチ:

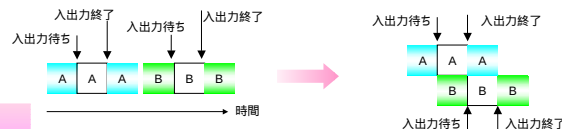
- プロセスにプロセッサを割り当てること
- 中断の際には、実行再開できるように中断された時点のCPUの状(コンテキスト)を保存する

Ichiro Satoh

プロセスの切り替えタイミング

- プロセスの実行が終了したとき
- 入出力の完了待ちや他のプロセスとの同期によりプロセスが実行を継続できなくなったとき
- 優先して実行すべきプロセスがあるとき
- 時分割処理においてそのプロセスに与えられた時間が経過したとき

他の実行可能なプロセスがあればその実行を開始する

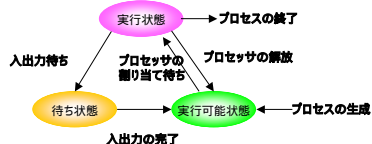


Ichiro Satoh

プロセスの状態

プロセスの実行上の状態:

- 実行状態 (running state)**
プロセッサが割り当てられている状態 (実際にプログラムを実行している)
- 実行可能状態 (ready state)**
プロセッサが割り当てられれば実行できる状態
- 待ち状態 (waiting state)**
入出力の完了待ちなどの何らかの事象を待っている状態

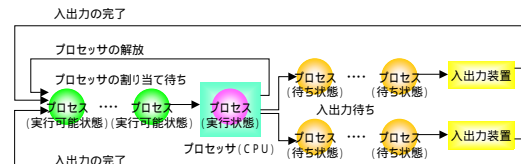


Ichiro Satoh

資源待ち行列

プロセッサ(CPU)だけでなくコンピュータ資源(入出力装置)は一度に一つのプロセスにしか対応できないことが多い。

- 実行可能状態プロセスは待ち行列に並びプロセッサの割り当てを待つ
- 待ち状態プロセスは入出力装置の割り当てまたは入出力の完了を待つ



Ichiro Satoh

プロセスの構造

OSはプロセス制御ブロック(PCB, Process Control Block)と呼ぶデータ構造によりプロセスに必要な情報を保持する。

プロセス識別子:

各プロセスに割り当てられる一意の番号

プロセス状態:

プロセス中断時にプロセスの状態(レジスタ内容、プログラムカウンタなど)を待避する領域

スケジューリング情報:

優先度や実行時間制約情報などを保持

計算資源情報:

プロセスが使用可能な計算資源一覧

PCB	
プロセス識別子	
プロセス状態	
スケジューリング情報	
計算資源利用情報	

Ichiro Satoh

スケジューリング

待ち行列中のプロセスからどのプロセスを選んで資源を割り当てるか

スケジューリングの目標

- スループット(単位時間あたりの処理量)の向上
- 応答時間の短縮
- 公平な割り当て
- 応答時間の予測可能性
- 所定時間内の処理
- 特定プロセスの優先的処理

すべての要求を満足するスケジューリングは存在しない

Ichiro Satoh

プロセスの切り替え方法

横取り可能スケジューリング(preemptive scheduling)

- プロセスの実行中に中断して他のプロセスの実行を許す
例: 所定時間が経過すると強制的に実行可能状態に移行

非横取り可能スケジューリング(non-preemptive scheduling)

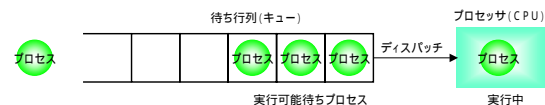
- プロセスの実行中の中断を許さない
例: プロセスの終了またはプロセス自身の中断により他のプロセスが実行可能

Ichiro Satoh

代表的なスケジューリング

どの順番でディスパッチ(プロセスにプロセッサを割り当てること)するか

- 到着順
- ラウンドロビン
- 処理時間順
- デッドライン時間順
- 優先度順
- 多重フィードバック



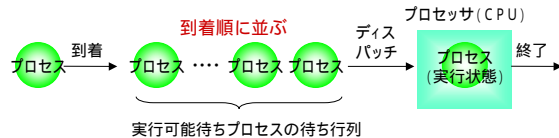
Ichiro Satoh

到着順スケジューリング

FCS (First Come First Service)

プロセスが到着した(新たに実行可能になった)プロセスから順にディスパッチ

- 処理が単純
- × プロセスの実行が長時間待たされる可能性



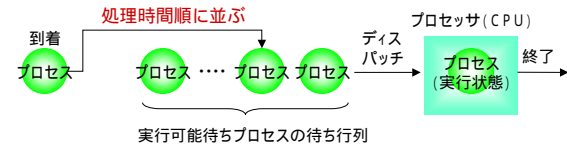
Ichiro Satoh

処理時間順スケジューリング

SPT (Shortest Processing Time First)

処理時間の短いプロセスから順にディスパッチ

- 処理時間の短いプロセスの待ち時間が短縮
- × プロセスの実行時間の予測は困難



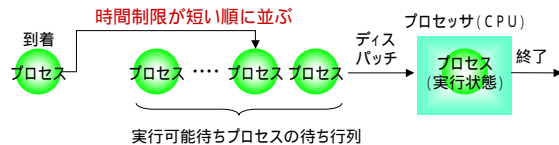
Ichiro Satoh

期限付きスケジューリング

Deadline Scheduling

実行完了までの時間制限が早いプロセスから順にディスパッチ

- 時間制限内に終了する可能性がある
- × 横取り可能時に実現が複雑になる



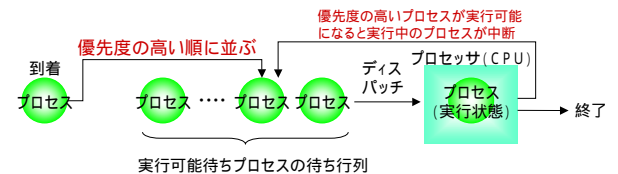
Ichiro Satoh

優先度順スケジューリング

Priority Dispatching

各プロセスに実行の優先度を与えて、優先度の高いプロセスから順に実行する

- 優先度の高いプロセスの実行完了が早くなる
- × 優先度の低いプロセスは待ち時間が長くなる



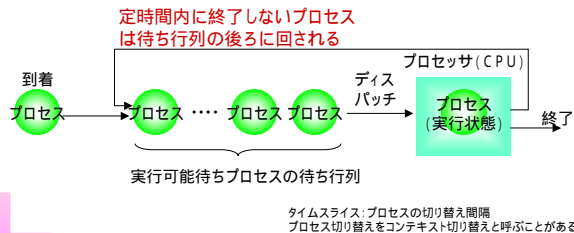
Ichiro Satoh

ラウンドロビンスケジューリング

RR (Round Robin)

プロセス実行開始後に所定時間が経過すると処理を中断して待ち行列の最後に回り、代わりに待ち行列の先頭のプロセスを実行

→ 横取り可能なスケジューリングの広く利用



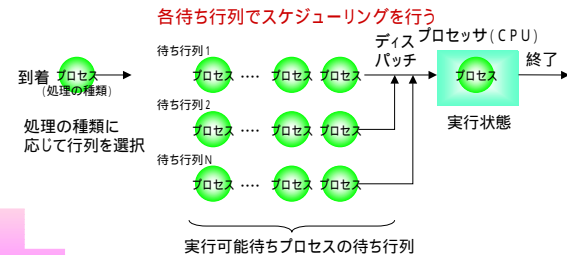
Ichiro Sato

多重レベル待ち行列スケジューリング

Multilevel Queue

複数の待ち行列を用意して、行列ごとに相違なスケジューリング方法を採用

→ 処理内容に応じたスケジューリングが可能



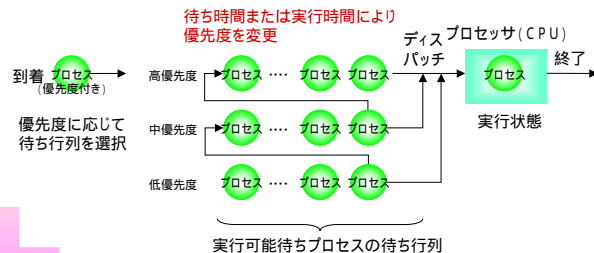
Ichiro Sato

多重レベルフィードバックスケジューリング

Multilevel Feedback Queue

優先度ごとに待ち行列を用意し、ある待ち行列で待ち時間が一定時間以上になると優先度を上げる

→ 優先度の低いプロセスの飢餓状態を回避

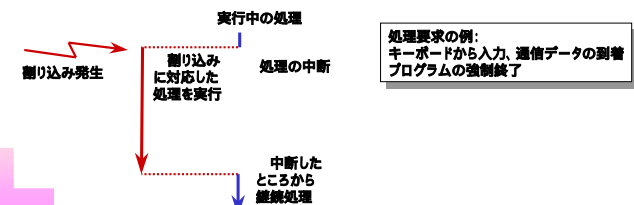


Ichiro Sato

割り込み

入出力完了などにプロセスの実行をその場で中断して、予め定義した別のプログラムを実行し、その後にもとのプロセスの処理を再開する。

- 1 プログラム処理中に何らかの割り込み要求が到着する
- 2 OSまたはハードウェアによりプログラムを一時的に中断する
- 3 処理要求のためのプログラムを実行する
- 4 一時中断したプログラムを再開する



Ichiro Sato