

システムプログラム

第14回

理工学部情報科学科

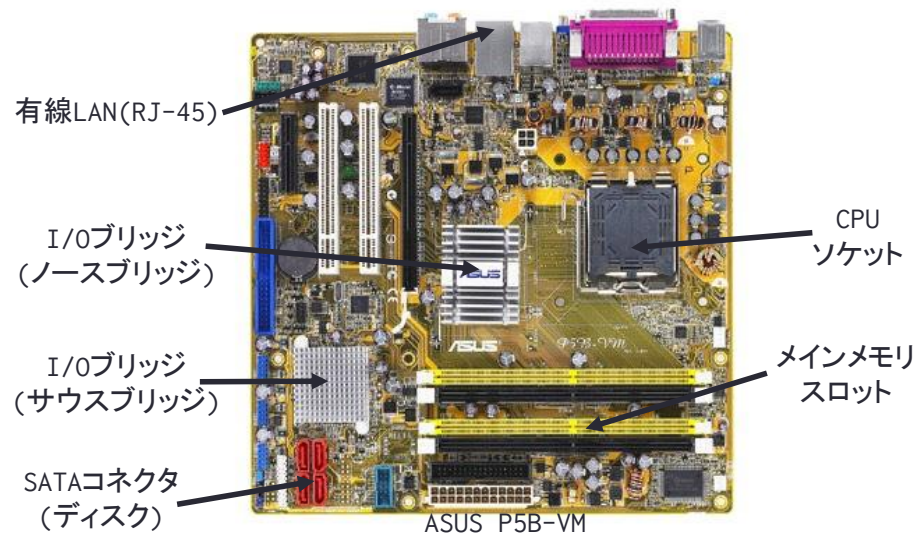
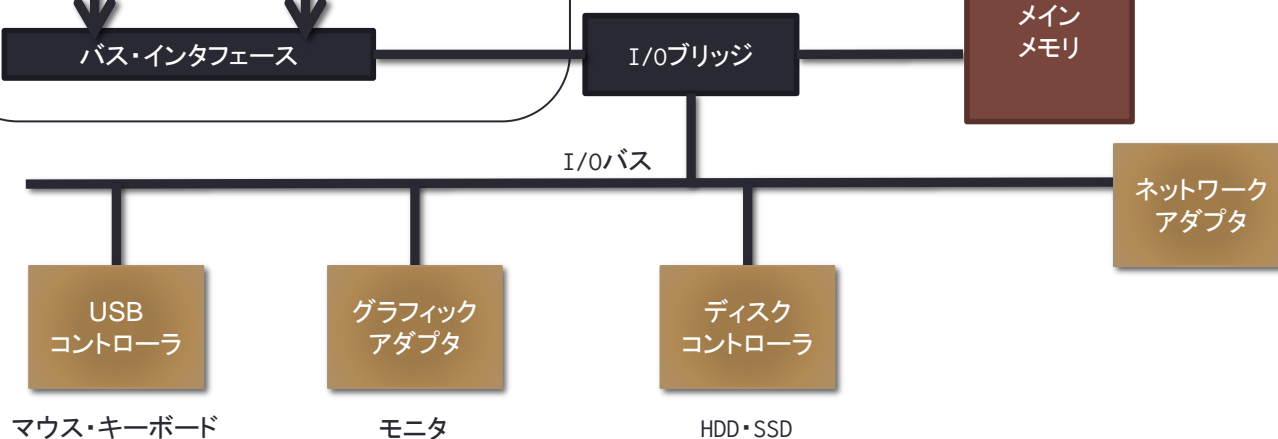
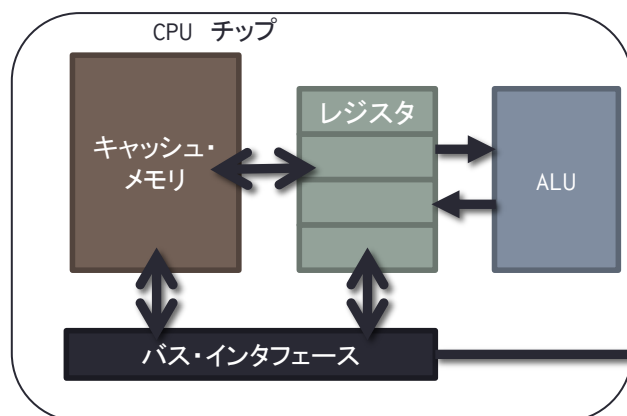
松澤 智史

本日の内容

- システムプログラム復習

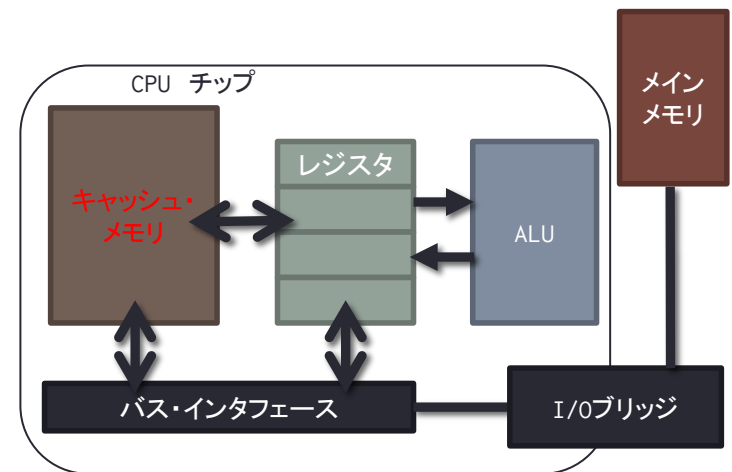
プログラムの実行

システムのハードウェア構成



キャッシュ・メモリ

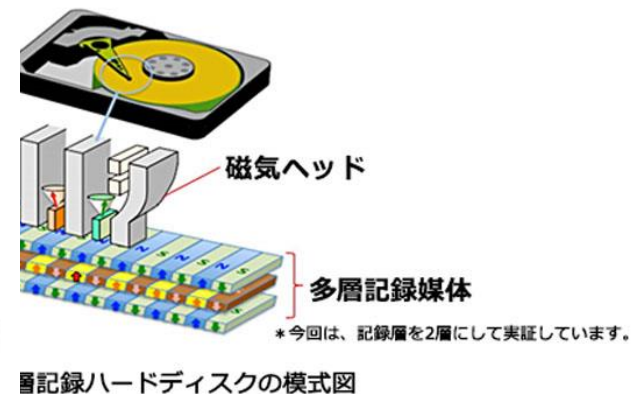
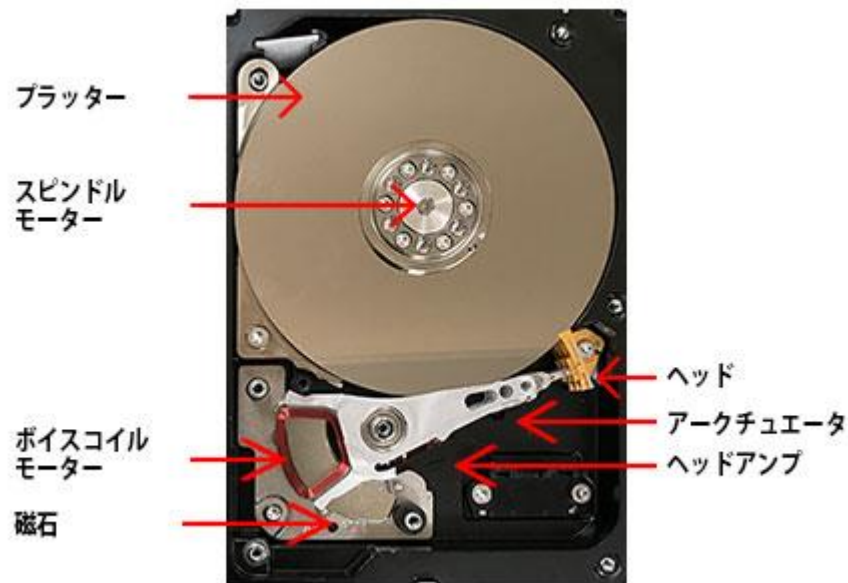
- 情報のある場所から別の場所に移す処理が多い
 - ディスクからレジスタを経由してメインメモリへ
 - メインメモリからレジスタへ
 - など
- レジスタの容量は64ビット×数十本であるが、メインメモリより100倍以上速く読み書きできる
 - メインメモリへの読み書きがネックとなる
 - プロセッサ-メモリギャップという
年々加速中
- CPUチップ内にメインメモリより高速な記憶媒体を用意する(キャッシュメモリ)



計算機内部の情報表現

- 現代の計算機は情報を2値の信号として格納している

例:HDD

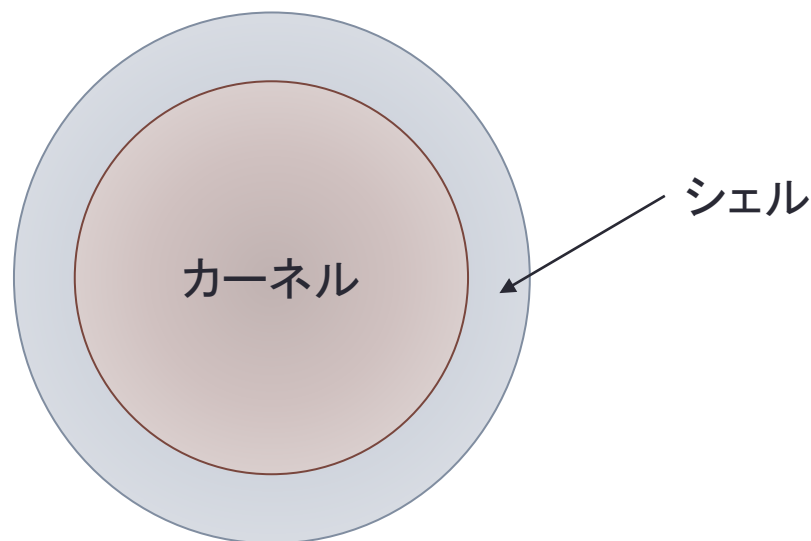


ビット

- 二つの数値に過ぎない「ビット」がデジタルの基礎
 - 10本の指を持つ人間は10進表現を使うことが一般的
 - 情報を格納して処理する機械は2進表現が効果的
- 複数のビットをグループ化し、様々なビットパターンに何かの**解釈**を与えることで、任意の有限集合の要素を表現
 - ビットは一つだけでは大した価値を持たない
 - 数値としての解釈
 - 文字としての解釈
 - 画像としての解釈
 - ブール値としての解釈 など

カーネル

- OSの中核となるソフトウェア
- ハードウェアを抽象化
 - ハードウェアとソフトウェアがやり取りできるようにする
 - ハードウェア: CPU, メモリ, ディスク, その他各種デバイス
- 機能
 - プロセス管理
 - メモリ管理
 - デバイス管理
 - ファイルシステム
 - ネットワーク
 - 他



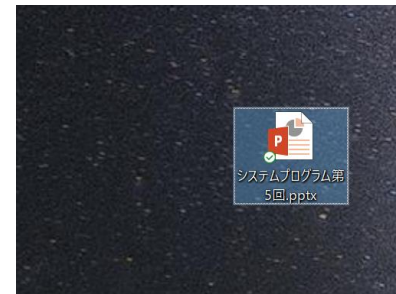
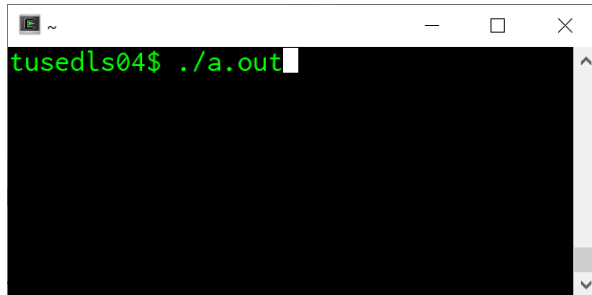
カーネルとシェル

- カーネル
 - プロセス管理
 - メモリ管理
 - ファイル管理, ファイルシステム
 - ネットワーク等のデバイス管理
 - etc
- シェル
 - ユーザとカーネルのインタフェース
 - インタプリタとして動作し, 対話型でユーザの入力を受理
 - 非対話型でも動作可能

様々なシェルの種類や特徴は第9回参照

プロセス

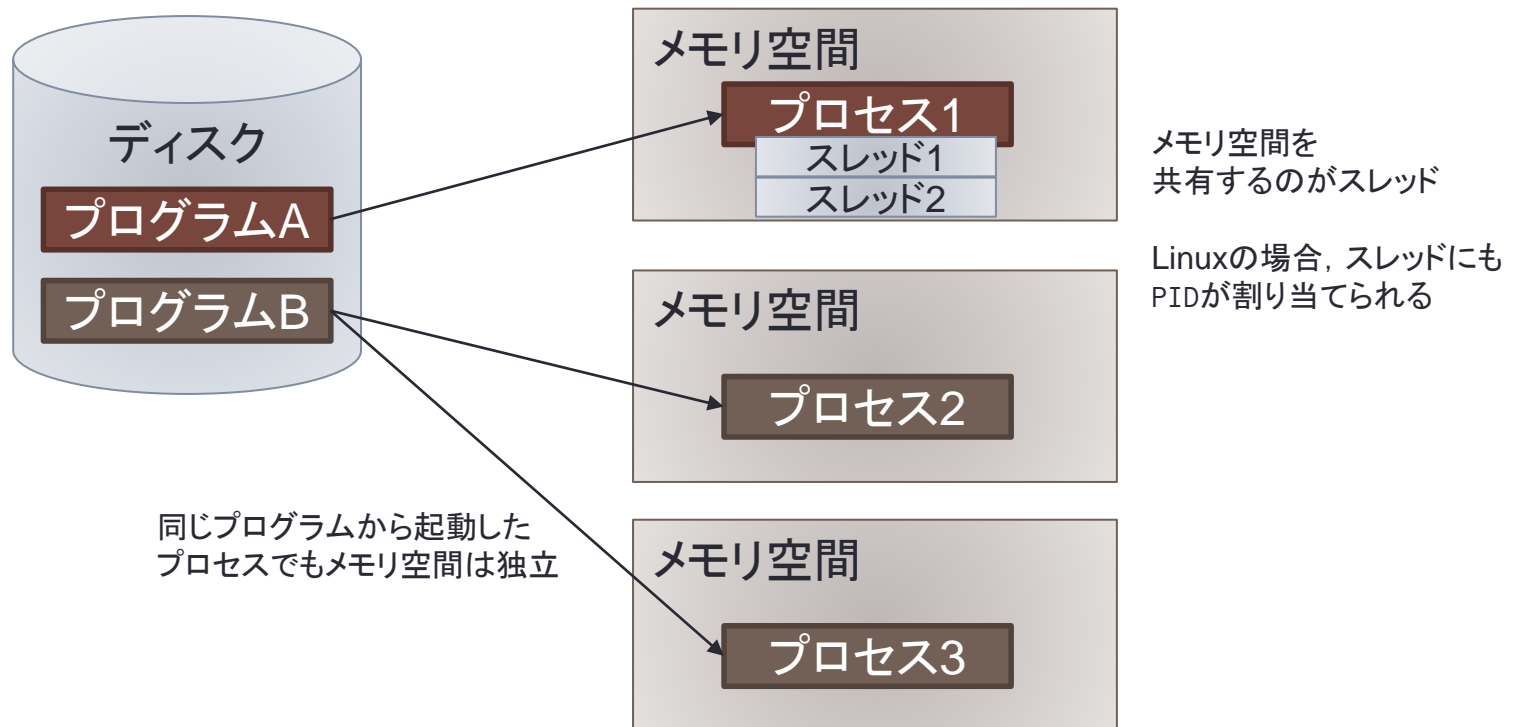
- プログラムの実行単位（タスク, ジョブなどと呼ばれることもある）
- プログラムとデータはメモリ上にあり, CPUによって逐次取り出して実行する
- 複数のプログラムを並行処理できる
 - 同じプログラムを並行処理はできない
 - 同じプログラムの平行動作はスレッド(Linuxカーネル2.4以降)で行える



コマンドラインから実行, ダブルクリックで実行
→プロセス起動

プロセス(タスク)管理

- プログラムが実行されるとカーネルはプロセスを生成する
- カーネルはプログラムを主メモリに読み込み、独立したメモリ空間を割り当てる
- プロセスにはプロセスID(PID)という管理番号が割り当てられる



並行処理

タスク(プロセス)

- Webブラウジング(Chrome, Safari等)
- 音楽再生
- ウイルススキャン

など

これらのプロセスの実行を適宜切り替えて、
同時に動いているように見せかける

余談:

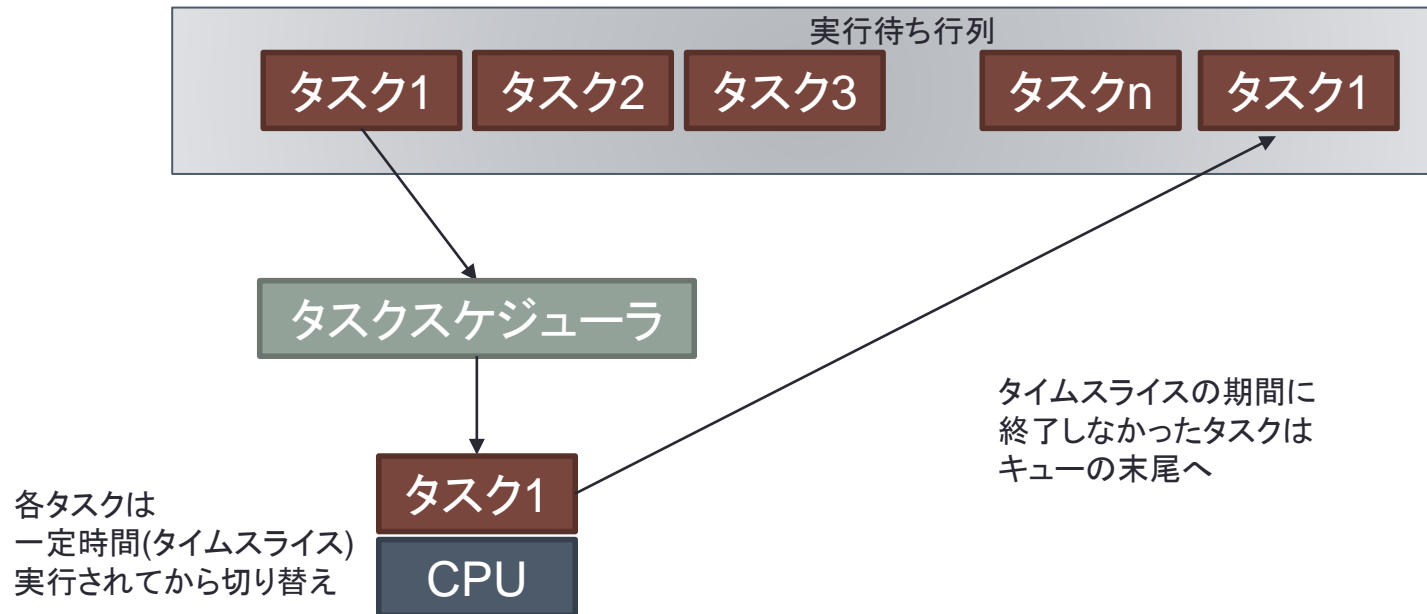
プロセスは計算機上でプログラムを動かす実行単位
タスクは行うべき仕事という抽象的表現
計算機上でのタスクはプロセスと同義

タスクスケジューリング

行うべきこと

「どのタスク(プロセス)」に「どのぐらいの時間実行」させて切り替えるか

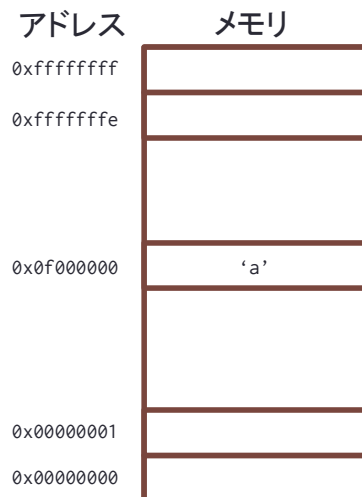
最も原始的な方法: ラウンドロビン



近年のスケジューリングアルゴリズムは第5回参照

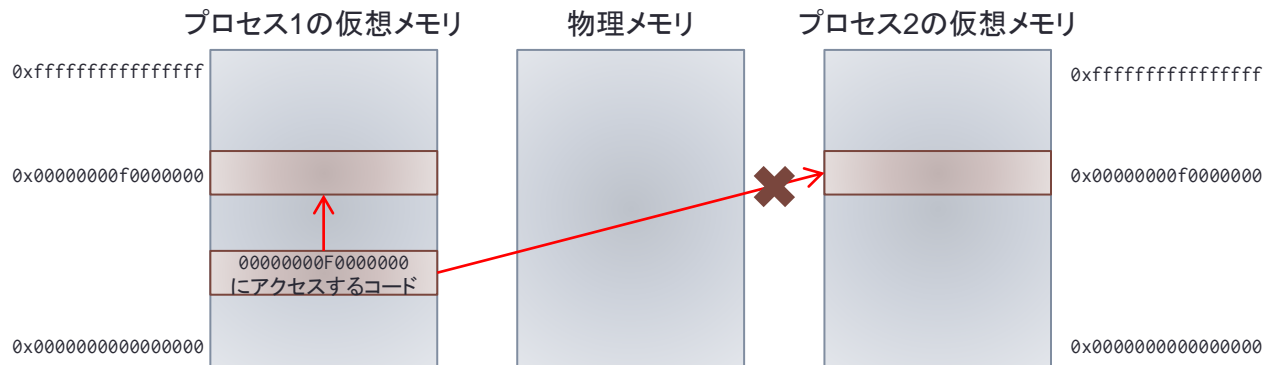
計算機のメモリ

- 1バイトごとにアドレスが割り当てられている
- 扱える上限がある
 - 32ビットのOSの場合は, $2^{32} \div \text{約40億バイト} \div 4\text{GB}$
 - 64ビットのOS(とCPU)の場合は 2^{64} バイト



仮想メモリと仮想アドレス空間

- カーネルは複数のプロセスを稼働する
- カーネルは各プロセスに独立したメモリ空間を提供する
 - 独立したメモリ空間は**仮想メモリ**と呼ばれる
 - 仮想メモリには**仮想アドレス空間**が割り当てられる
 - 仮想アドレスは物理メモリのアドレスとは関係ない

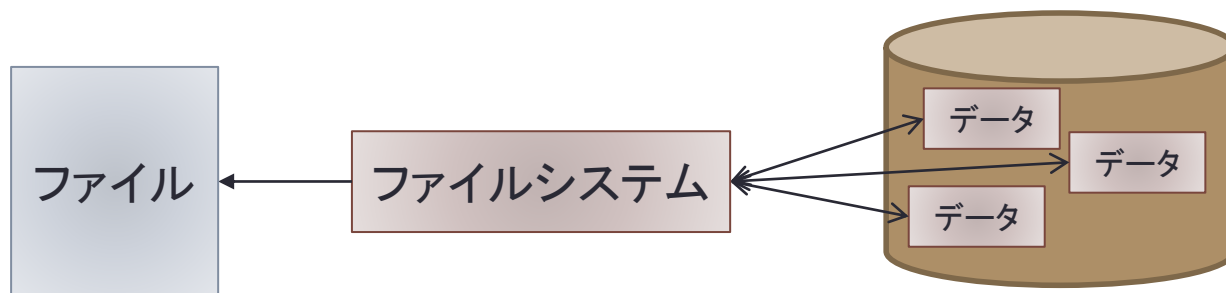


仮想アドレスの割り当て方
割り当てる領域の種類などは第6回参照

ファイルシステムの概要

- ファイルというインタフェースをユーザやアプリケーションに提供するシステム
- 記憶装置にデータがどのように記録されているのかを意識することなくデータをファイルという仮想的な容器に格納・取り出しが可能
- ファイルの入れ物である「ディレクトリ」を階層的に配置
- ファイルの管理性の向上
- 記録データのセキュリティの確保

ファイルシステムの種類や特徴・管理の仕方などは第7回参照



OSのネットワーク管理

メモリを共有しない対象とのデータ共有

- 他のプロセスとの通信
 - Inter Process Communication(IPC)
- 外部端末との通信
 - TCP/IPプロトコルスタック
 - AppleTalkプロトコルスタック
 - SMB/CIFSプロトコルスタック
 - etc

共有メモリの通信は主に並列処理

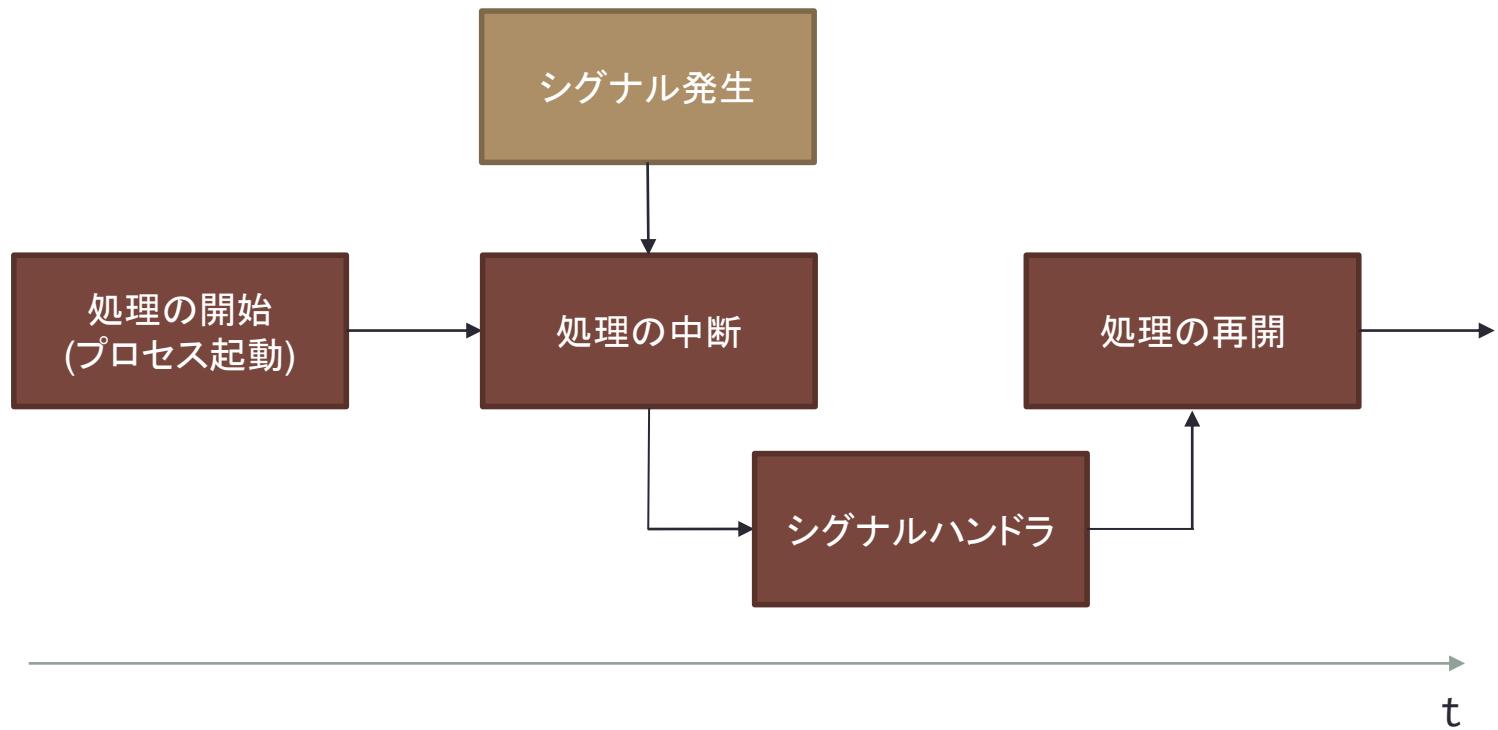
OSでサポートすべき階層



シグナル

- 非同期イベントをプロセスに伝える仕組み
- プログラム実行中の処理に割り込んで別の処理を行わせる
 - 割り込み処理ともいう
- Ctrl-c, Ctrl-zなどの入力もシグナルを発生させる

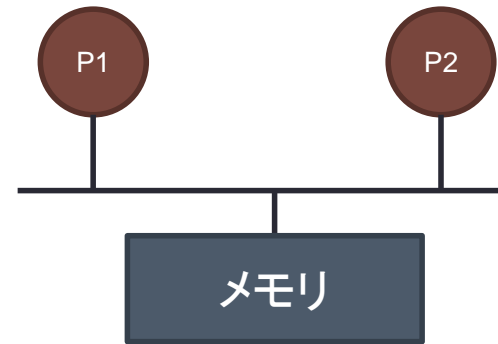
シグナルの動作



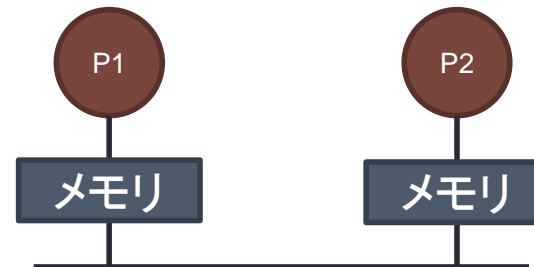
シグナルの種類や使い方などは第13回参照

マルチプロセッサ

- 共有メモリ型マルチプロセッサ
 - どのプロセッサからも同様にアクセス可能なメモリを持つ



- メッセージパッシング型マルチプロセッサ
 - 共有メモリを持たず
メッセージ交換で処理を行う



並列計算の可能性と問題点

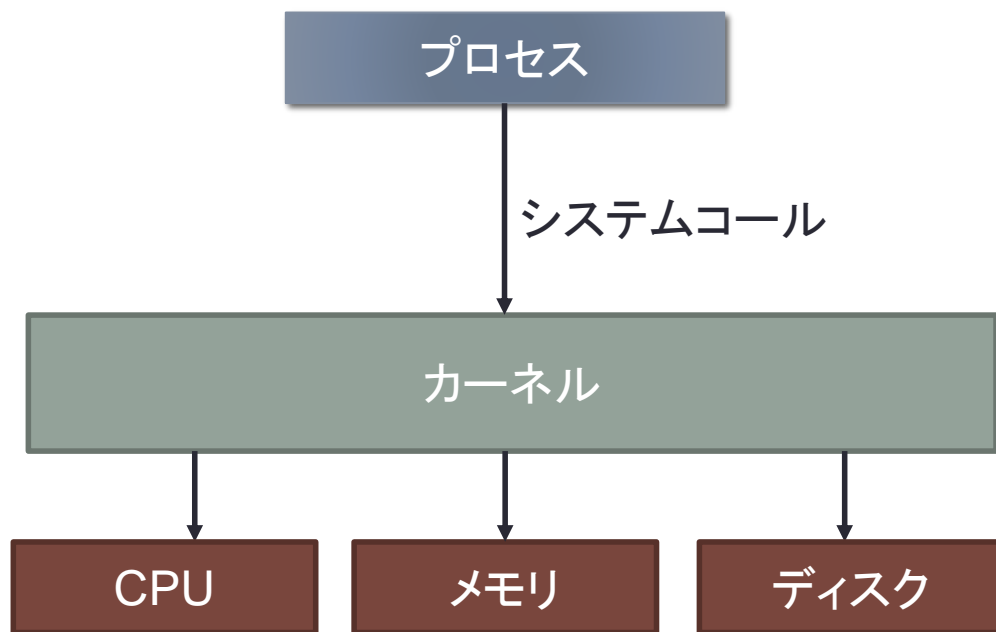
- 理想的には、プロセッサの数に比例した性能向上が可能
 - 理想的な環境 = 複数のプロセッサが100%の性能を発揮する
- 複数のプロセッサ性能を100%使い切るのは難しい
 - 他のプロセッサの結果に依存する計算の排除
 - 各プロセッサの仕事を均等に割り当てる工夫
- 並列化に必要な処理(オーバーヘッド)が発生する
 - 並列化の恩恵をオーバーヘッドが上回るケースもある

複数のプロセッサで均等に処理できる部分の割合を
プログラムの**並列度**と呼ぶ

並列計算の理論・サンプル・
実行方法などは第11回参照

システムコール

- カーネルはCPUやメモリ, 各種デバイスの管理をする
- カーネルの機能の実行依頼をシステムコールと呼ぶ



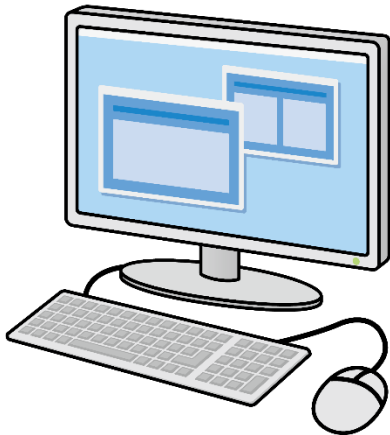
システムコール

- アプリケーション開発者が直接システムコールを扱うことは稀
 - ※間接的には必ず使うことになる
- 直接扱う例
 - 処理速度を限界まであげるようなシステム開発
 - デバイスドライバなどのハードウェアを直接扱うシステム開発
- 多くのプログラミング言語やツールは、システムコールを扱いやすくするための機能やライブラリを提供

システムコールの詳細や種類などは第10回参照

X Window System

- GUI環境を実現するシステム
- Xサーバと Xクライアントに分割



Xサーバ



Xクライアント

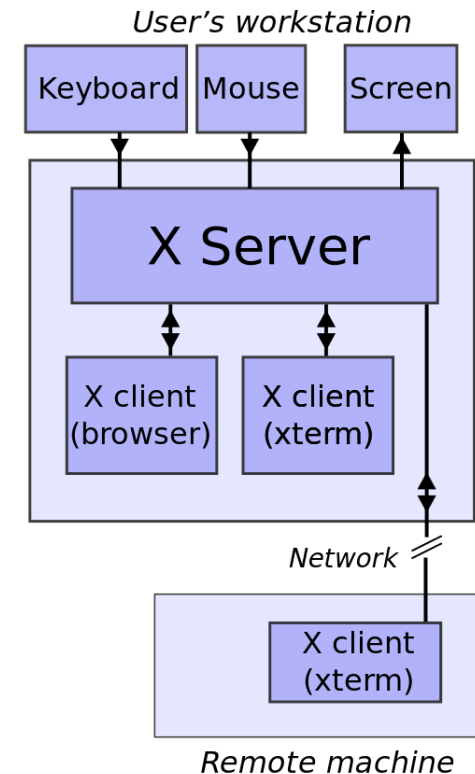
X Window System

- Xサーバ

- マウス, キーボード, タッチパネル等からの入力をXクライアントに伝える
- Xクライアントからの指示にしたがってディスプレイに描画する

- Xクライアント

- ウィンドウ表示される端末などが該当する



最後に

OSの歴史と今後(OSの原点)

初期のコンピュータ

- 用途別開発
 - 事務用コンピュータ, 科学計算用コンピュータ
- コンピュータのユーザはすべて自身でプログラム開発
 - コンピュータの使う人は専門家限定



どのコンピュータでも使う処理のプログラムが提供開始

- 当初は入出力の制御プログラム
- 計算機を効率的に使う工夫
- 使いやすさ向上



OSの原点が形成

OSの歴史と今後(時代はパーソナルへ)

1960年代, 計算機は非常に高価

- 多くのユーザーが同時に利用するシステムの要望大
- Multics:オンライン・マルチユーザー・システムの開発



1969年, Multicsの機能を整理し改善したOS:Unics

- 後にUNIXに改名, 大学や企業に販売(ソースコードつき)

※このあたりからパーソナルが意識される

OSの歴史と今後(主要3OS)

Windows

- IBMがパーソナルコンピュータ(IBM PC)の開発に着手
- ソフトウェアの外注先としてマイクロソフト社を選定
- MS-DOS誕生(CUIベース)
- 1985年にグラフィカルなUI

MacOS

- 1984年, アップル社の Macintosh に搭載されていたOS
- GUIを備えた最初のパーソナルコンピュータ用OS

Linux

- リーナス・トーバルズがUNIXを参考に0から作成したOS
- オープンソース

OSの歴史と今後(これからの時代)

- Android iOSなどの新OSが登場
- デバイスが変わればOSも変化
 - キーボード/マウス→タッチパネル
→音声・ジェスチャ・VR・空間ディスプレイ？
- 市場の変化
 - 集団→パーソナル→次は？
- 技術革新
 - AI開発用OS？

計算機分野の歴史は他の分野で繰り返すこともある
新時代を皆さんで創り上げていってください

全14回お疲れさまでした

毎回色々詰込みましたが、
講義1回分でも、余談の1つでもお話した内容が
今後の皆さんの役に立てば幸いです