

オペレーティングシステム

第1回

オペレーティングシステムの基本概念

<http://www.info.kindai.ac.jp/OS>

E館3階E-331 内線5459

takasi-i@info.kindai.ac.jp

本科目の内容

- 本科目で学ぶこと
 - オペレーティングシステム(OS)の基本概念
 - OSの役割・目的
 - プロセス・スレッドの生成・管理
 - スケジューリング
 - プロセス間通信・デッドロック
 - メモリ管理
 - 割込み
 - 入出力の制御

成績について

評価基準

オンライン課題 (毎週)	30%
オンライン試験 (第15週)	70%

● 無届欠席禁止

- やむを得ず欠席した場合は翌週までに欠席届を提出すること
- 無届欠席が複数回ある場合は試験の点数に関わりなく不受となる

オンライン授業では GoogleClassroom から出席カードが提出されれば出席とします

昨年度の受講状況

学年	コース	受講者数	合格	不可	不受	合格率
2	システム	89	85	1	3	97%
2	メディア	4	3	1	0	75%
3	システム	2	2	0	0	100%
3	メディア	5	5	0	0	100%
4	メディア	1	1	0	0	100%

※全出席し、全レポートを〆切までに提出して
不可になった受講生はいない



ToDo チェックが必要な課題 カレンダー

2022-科学技術の発展...
法学部

2022-オペレーティン...
理工学部 情報学科 情報システム科
制御プログラム

情報学部志願者のた...
近畿大学情報学部

基礎ゼミ1 (Aグルー...
Aグループ

2022年度才
理工学部情報学科の

<https://classroom.google.com/h>

ToDo チェックが必要な課題 カレンダー

クラスに参加
クラスを作成

2022-科学技術の発展...
法学部

2022-オペレーティン...
理工学部 情報学科 情報システム科
制御プログラム

情報学部志願者のた...
近畿大学情報学部

基礎ゼミ1 (Aグルー...
Aグループ

2022年度才
理工学部情報学科の

<https://classroom.google.com/h>

現在、次のメールアドレスでログインしています



アカウントを切り替える

クラスコード

教師にクラスコードを聞いてこちらに入力してください。

クラスコード

クラスコードを使用してログインするには

- 承認済みアカウントを使用します
- 5~7個の文字と数字で構成され、スペースや記号を含まないクラスコードを使用してください

クラスに参加できない場合は、[ヘルプセンターの記事](#)を確認してください

bgbto2b

ToDo チェックが必要な課題 カレンダー

2022-科学技術の発展...
法学部

2022-オペレーティン...
理工学部 情報学科 情報システムコース
制御プログラム

情報学部志願者のた...
近畿大学情報学部

基礎ゼミ1 (Aグルー...
Aグループ

2022年度オリエンテ...
理工学部情報学科の在学生

基礎ゼミ1 (再履修ク...

2022-オペレーティングシステム
理工学部 情報学科 情報システムコース 2年

OS

仮想計算機

資源管理者

カスタマイズ

Meet

リンクを生成

クラスコード

bgbto2b

期限間近

提出期限の近い課題はありません

すべて表示

保存済みのお知らせ (5件)

クラスへの連絡事項を入力

石水隆
8月27日

出席カードを提出してください (9/12) <https://forms.gle/rjRvuSzQhy6qKnGs6>

クラスのコメントを追加...

石水隆 さんが新しい資料を投稿しました: 出席カード (9/12)

出席カード (OS) 第1回 9/12

.kindai.ac.jp アカウントを切り替える

このフォームを送信すると、メールアドレスが記録されます

***必須**

あなたの氏名を入力してください。 *

回答を入力

あなたの学籍番号を入力してください。 (例 : 2110370999) 省略形は使用しないでください。 *

回答を入力

回答のコピーを自分宛に送信する

送信

フォームをクリア

Google フォームでパスワードを送信しないでください。



このページは2022年度

https://www.info.kindai.ac.jp/OS/

お知らせ

• 出席について

単位取得には原則として全ての授業に出席する必要があります。やむを得ず欠席する場合はその翌週までに必ず欠席届を出してください。欠席届無しの欠席が複数回ある場合は履修の意思無しと見做して不受扱いにします。

オンライン授業では、当日 [GoogleClassroom](#) から出席カードが提出がされていれば出席扱いにします。

• 課題について

単位取得には原則として全ての課題テストを受講する必要があります。正当な理由のある場合を除いて原則として締切を過ぎた受講は認めません。

課題は原則として授業当日に提出し、授業時間終了までにできなかった場合のみ、その翌週までに提出するようにしてください。分からない場合は質問も受け付けますので、少なくとも16時30分までは解く努力をしてください。

講義資料

- 第1回 オペレーティングシステムの基本概念 (9/12) [パワーポイント](#) [PDF](#) [ノート用PDF](#) (8/28 update)
- 第2回 割り込みとOSの構成 (9/26) [パワーポイント](#) [PDF](#) [ノート用PDF](#) (8/28 update)
- 第3回 プロセスの管理とスケジューリング (10/3) [パワーポイント](#) [PDF](#) [ノート用PDF](#) (8/28 update)

課題テスト

課題テストは [GoogleClassroom](#) 上で行います。GoogleClassroom で「オペレーティングシステム」⇒「授業」⇒その回の「課題」と辿って受講してください。出席を兼ねていますので、必ずテストを受けてください。また、単位取得には全ての課題テストの受講が必要です。

+ 作成

Google カレンダー クラスのドライブ フォルダ

すべてのトピック

第6回 プロセス間通...

第5回 プロセスの相...

第4回 プロセス生成...

第3回 プロセス管理...

第2回 割り込みとO...

第1回 OSの概要

第6回 プロセス間通信

- 第6回 講義資料 下書き
- 第6回 課題 各週課題 投稿予定: 10月25日 8:00
- 参考Javaプログラム 下書き
- 出席カード (第6回) 投稿予定: 10月31日 8:00

第5回 プロセスの相互排除

- 第5回 講義資料 下書き
- 第5回 課題 各週課題 投稿予定: 10月18日 8:00
- 参考Javaプログラム 下書き



第3回 課題 各週課題 投稿予定: 9月27日 8:00

出席カード (10/3) 投稿予定: 10月3日 8:00

第2回 割り込みとOSの構成

第2回 講義資料 投稿日: 8月27日

第2回 課題 各週課題 投稿予定: 9月13日 8:00

出席カード (9/26) 投稿予定: 9月26日 8:00

第1回 OSの概要

第1回 講義資料 投稿日: 8月27日

第1回 課題 各週課題 期限: 9月26日

出席カード (9/12) 投稿日: 8月27日



第1回 OSの概要



第1回 講義資料

投稿日: 8月27日

OS01.pptx: パワーポイントファイル
OS01.pdf: pdf ファイル
OS01note.pdf: ノート用 pdf ファイル
OS01.mp4: 動画ファイル (178MB, 50分)
introduce.pdf: 補足資料



OS01.pptx
PowerPoint



OS01.pdf
PDF



OS01note.pdf
PDF



OS01.mp4
動画

資料を表示



第1回 課題

各週課題

期限: 9月26日



出席カード (0/10)

投稿日: 8月27日

あらまし

- オペレーティングシステム(OS)とは
- OSの役割
- OSの歴史
- OSの構成

計算機システムの要素

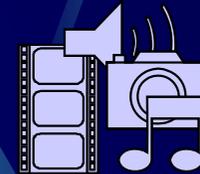
● ハードウェア

- 基本的な計算資源を供給する
 - CPU, 主記憶, IO装置等



● アプリケーションプログラム(AP)

- システム資源を利用してユーザの問題を解く
 - コンパイラ, データベースシステム, ビデオゲーム, ビジネスプログラム等



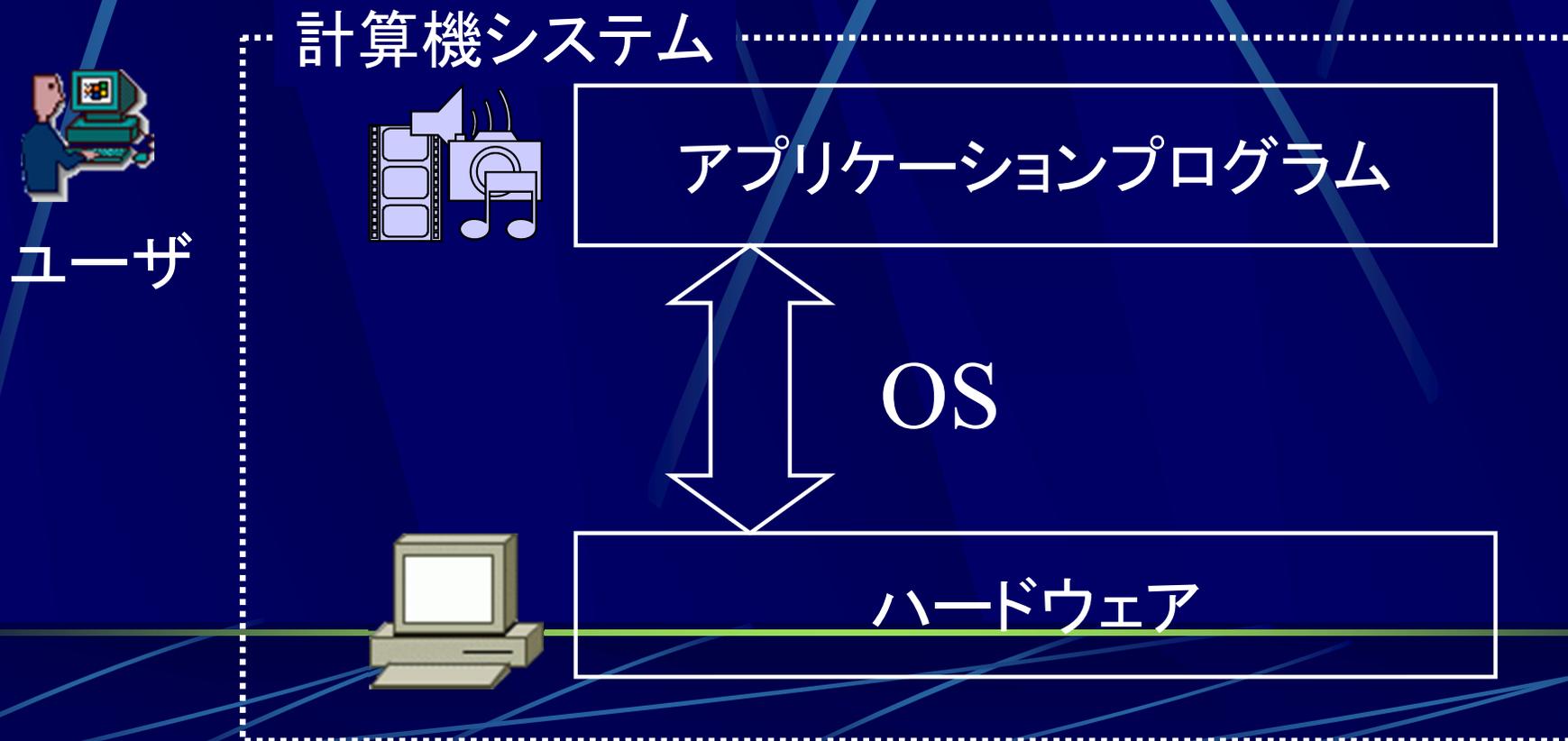
● ユーザ

- 計算機を利用して問題を解く
 - 人間, マシン, 他の計算機等



オペレーティングシステムとは

- ユーザとハードウェアの仲介の役割をするプログラム

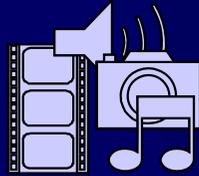


OSの概念図

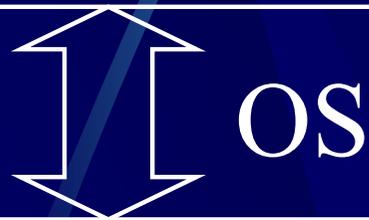
ユーザ



ウェブブラウザ テキストエディタ データベース



アプリケーションプログラム



ハードウェア

演算装置
記憶装置
入出力装置

OSの役割

資源管理者

OS

仮想計算機

制御プログラム



OSの役割: 資源管理者

(resource manager, resource allocator)

● 資源

- ハードウェア : CPU, メモリ, IO装置, ハードディスク
- ソフトウェア : プログラム, データ等

● 目的

- 共有資源の状態を維持
- APからのアクセスを管理, 資源を適正に利用させる

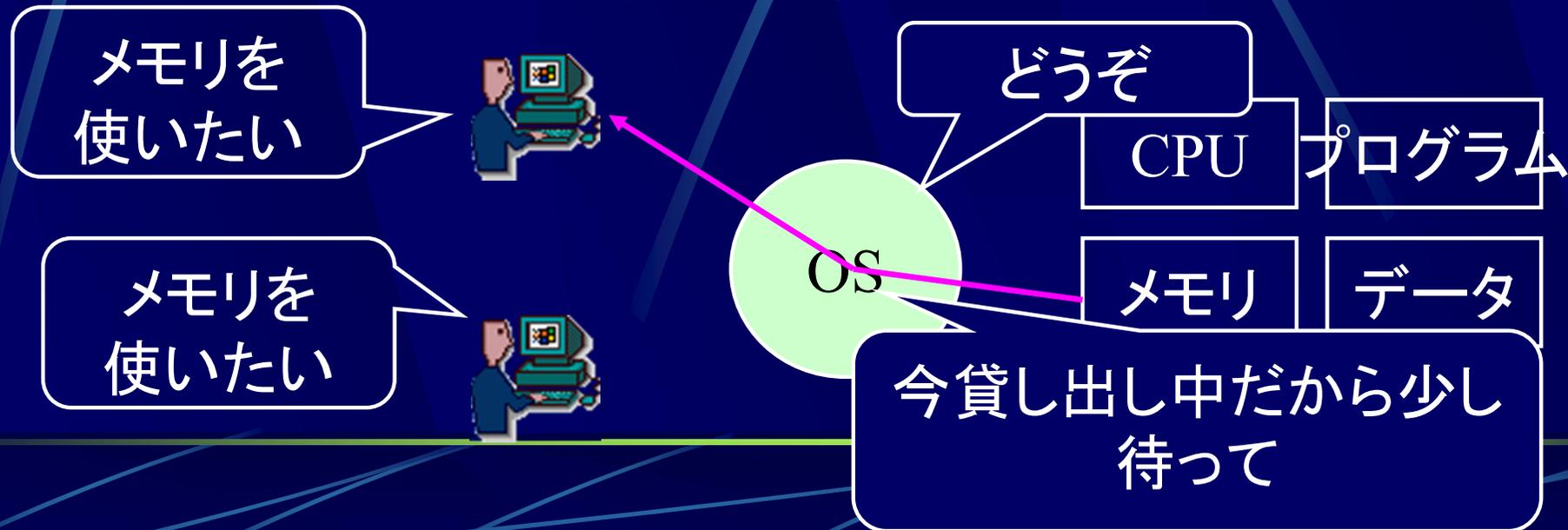
● 要求

- 効率
 - 公平性
- トレードオフ

公平性を追求すると
効率が下がる
逆も同じ

OSの役割：資源管理者

- 資源管理(resource manager)
 - ハードウェア資源の管理
- 情報管理(information manager)
 - ソフトウェア資源の管理



OSの役割：資源管理者



この4つのアプリケーションを使いたい

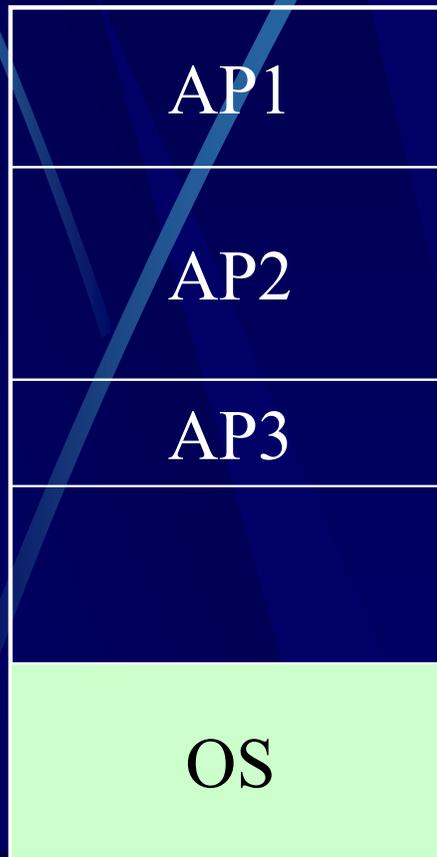
AP1

AP2

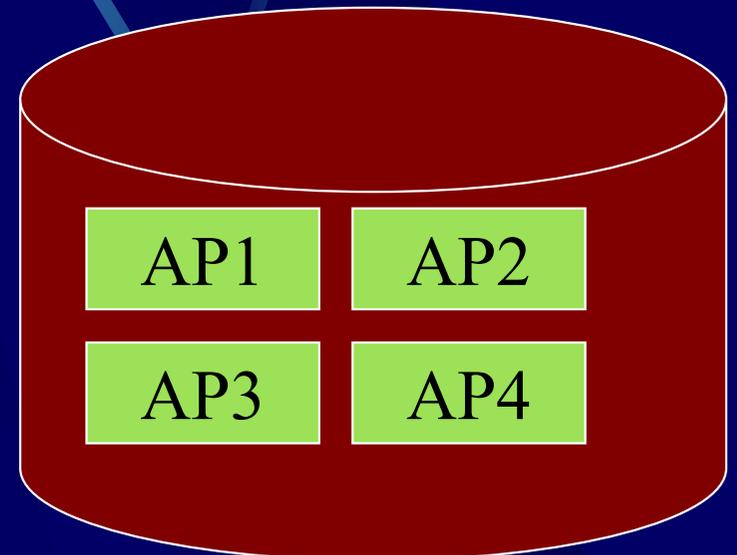
AP3

AP4

メモリ



ハードディスク

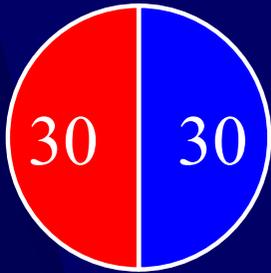


今は使わないからAP4はHDに置こう

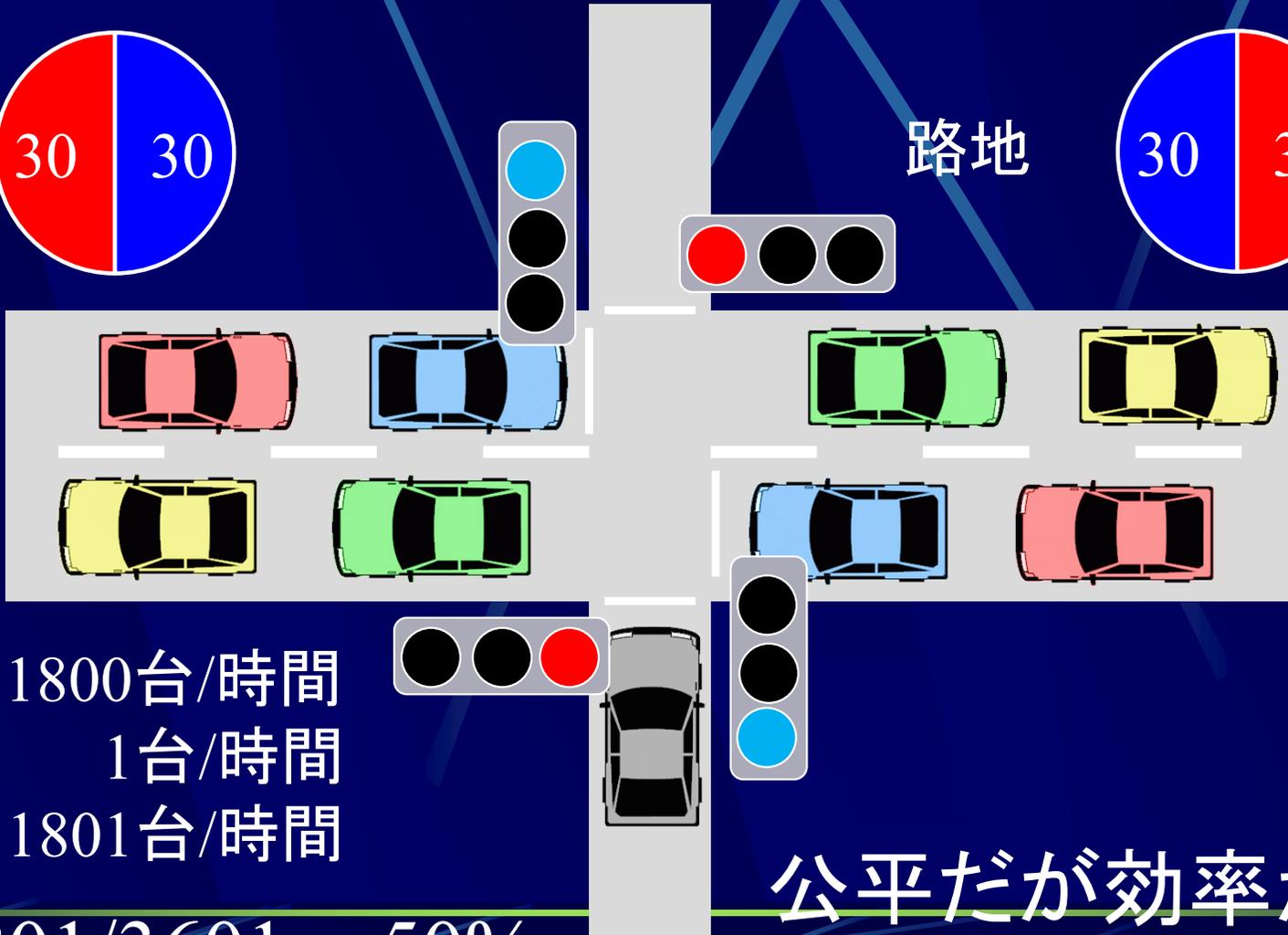
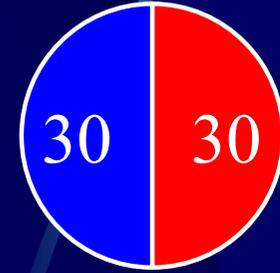
効率と公平性：公平性優先

赤と青を完全に半々にする

大通り



路地



大通り: 1800台/時間

路地 : 1台/時間

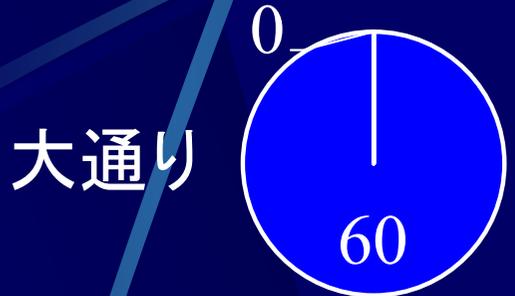
計 : 1801台/時間

効率: $1801/3601 = 50\%$

公平だが効率が悪い

効率と公平性：効率優先

大通りを常に青にする



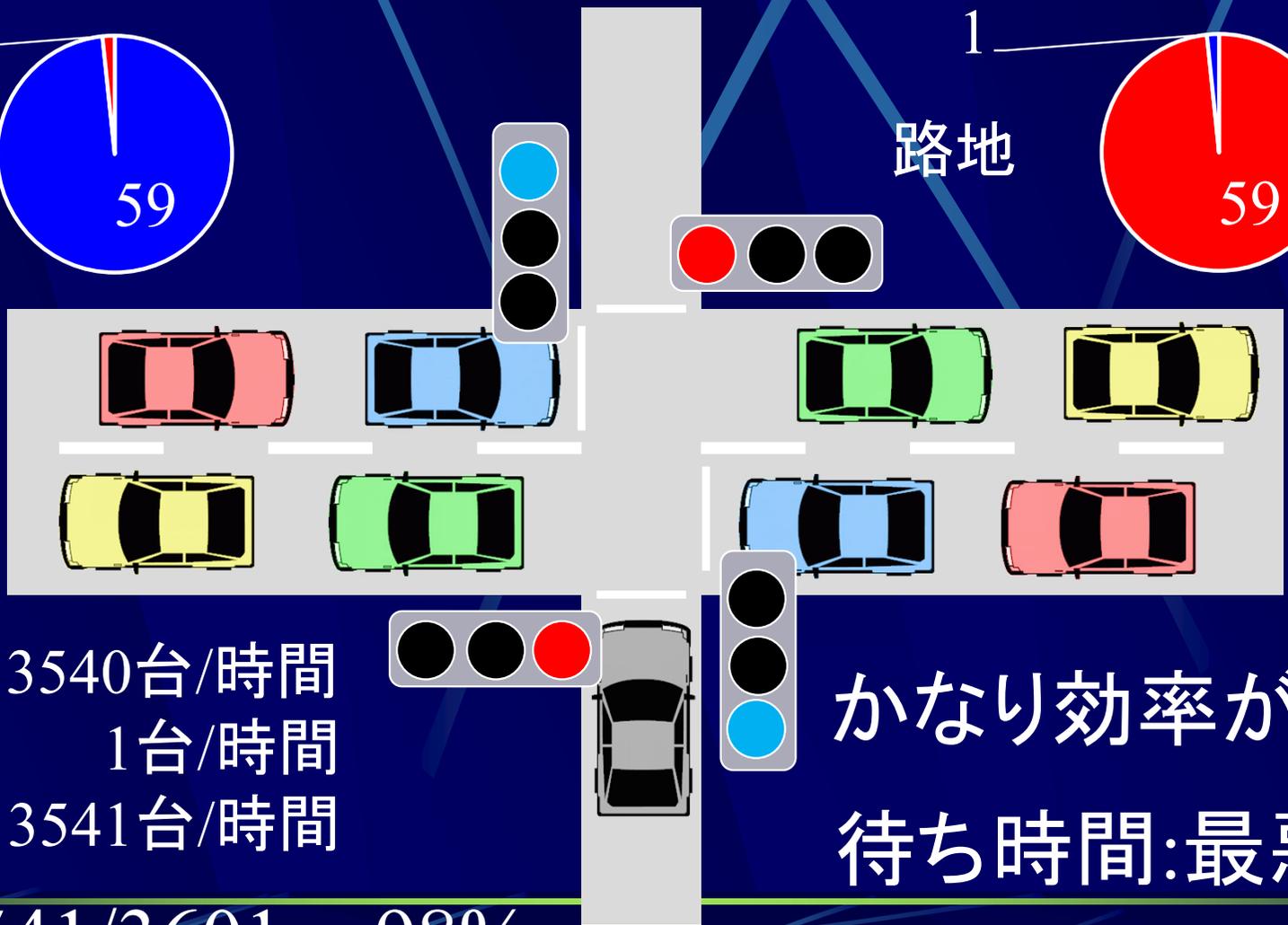
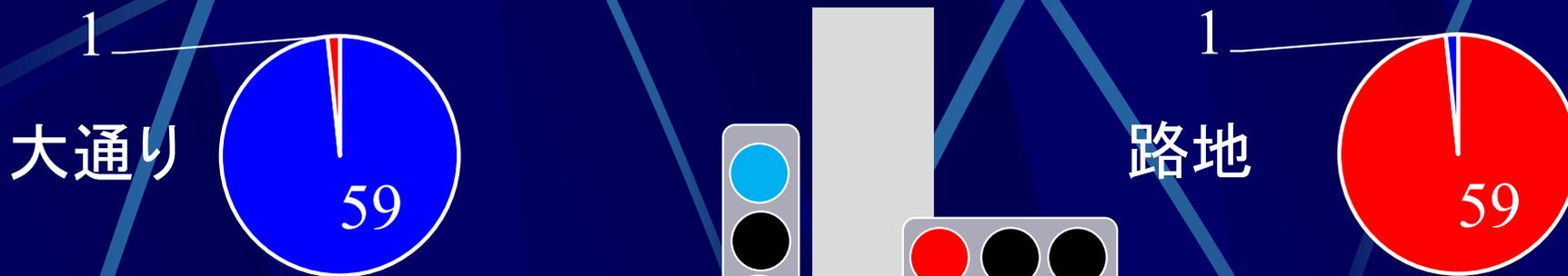
大通り: 3600台/時間
路地 : 0台/時間
計 : 3600台/時間

効率: $3600/3601 = 100\%$

路地の車が永久に動けない

効率と公平性：妥協案

路地を1時間に1分だけ青にする



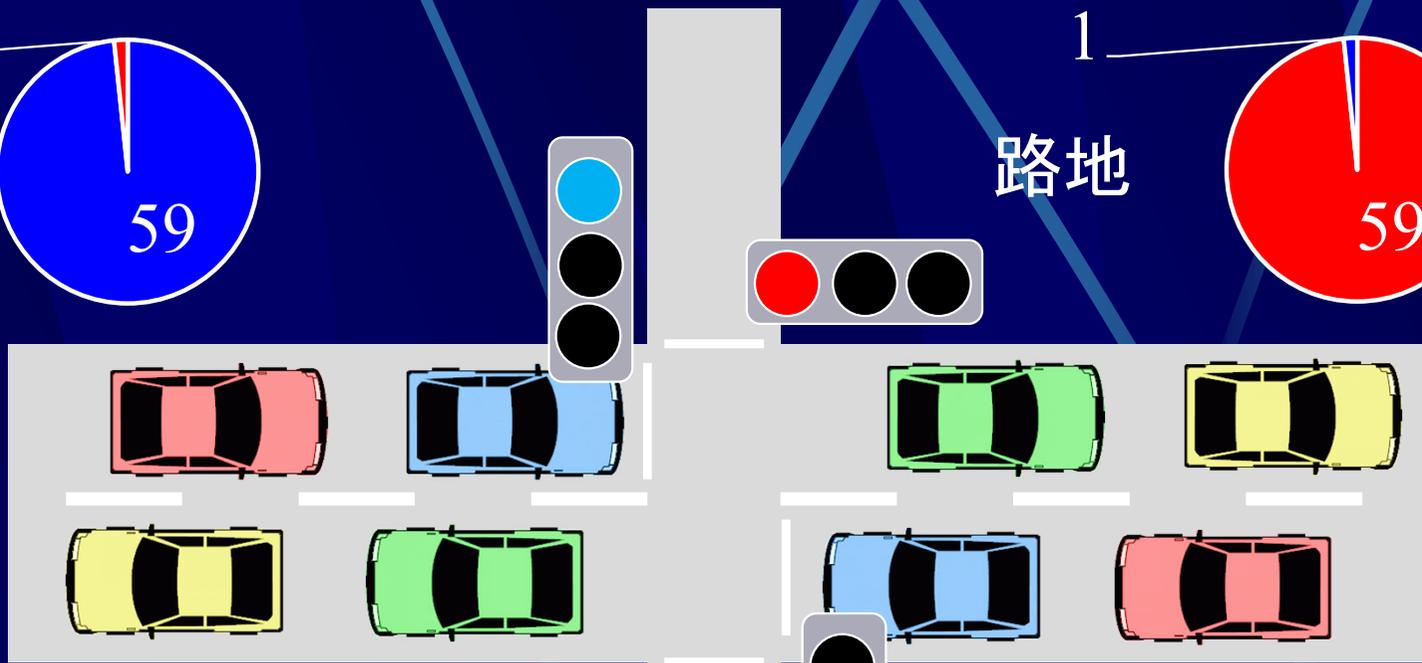
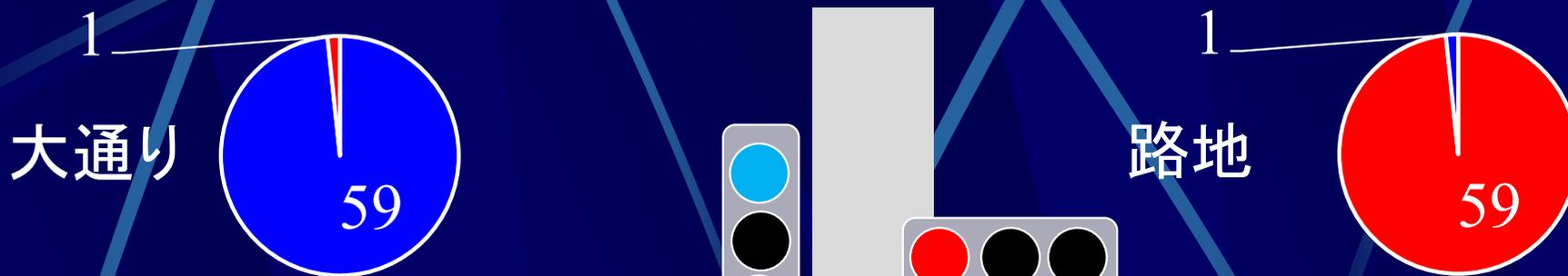
大通り: 3540台/時間
路地 : 1台/時間
計 : 3541台/時間

かなり効率がいい
待ち時間: 最悪59分

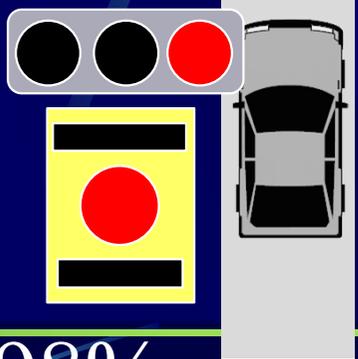
効率: $3541/3601 = 98\%$

効率と公平性：改良案

押しボタンが押されれば1分だけ青にする



大通り: 3540台/時間
路地 : 1台/時間
計 : 3541台/時間

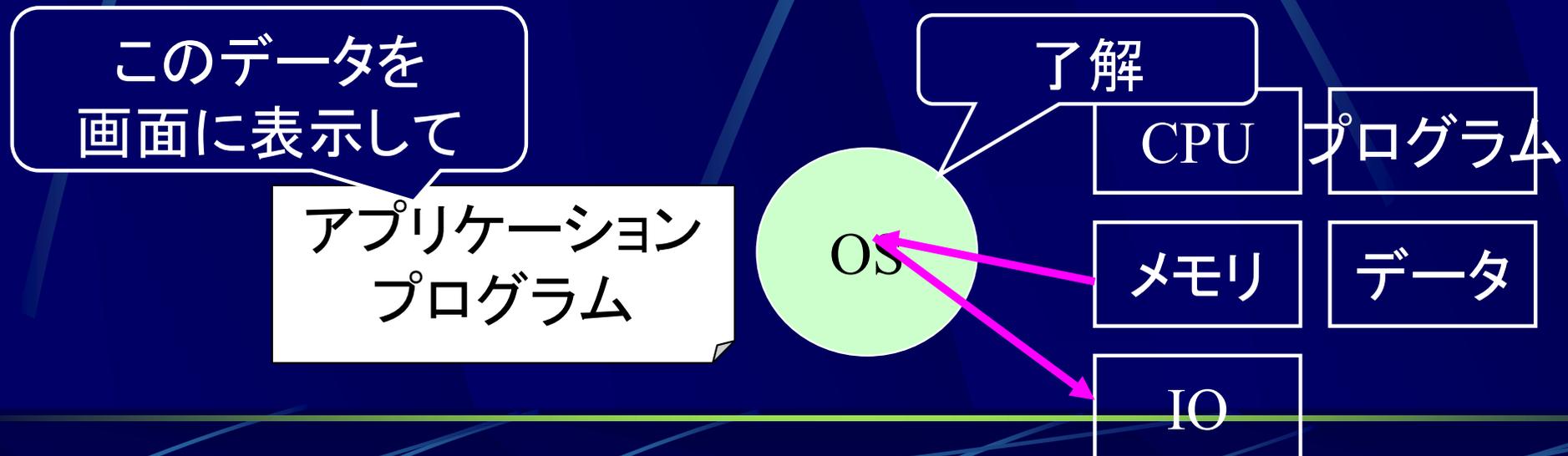


待ち時間: 最悪1分

効率: $3541/3601 = 98\%$

OSの役割：制御プログラム (control program)

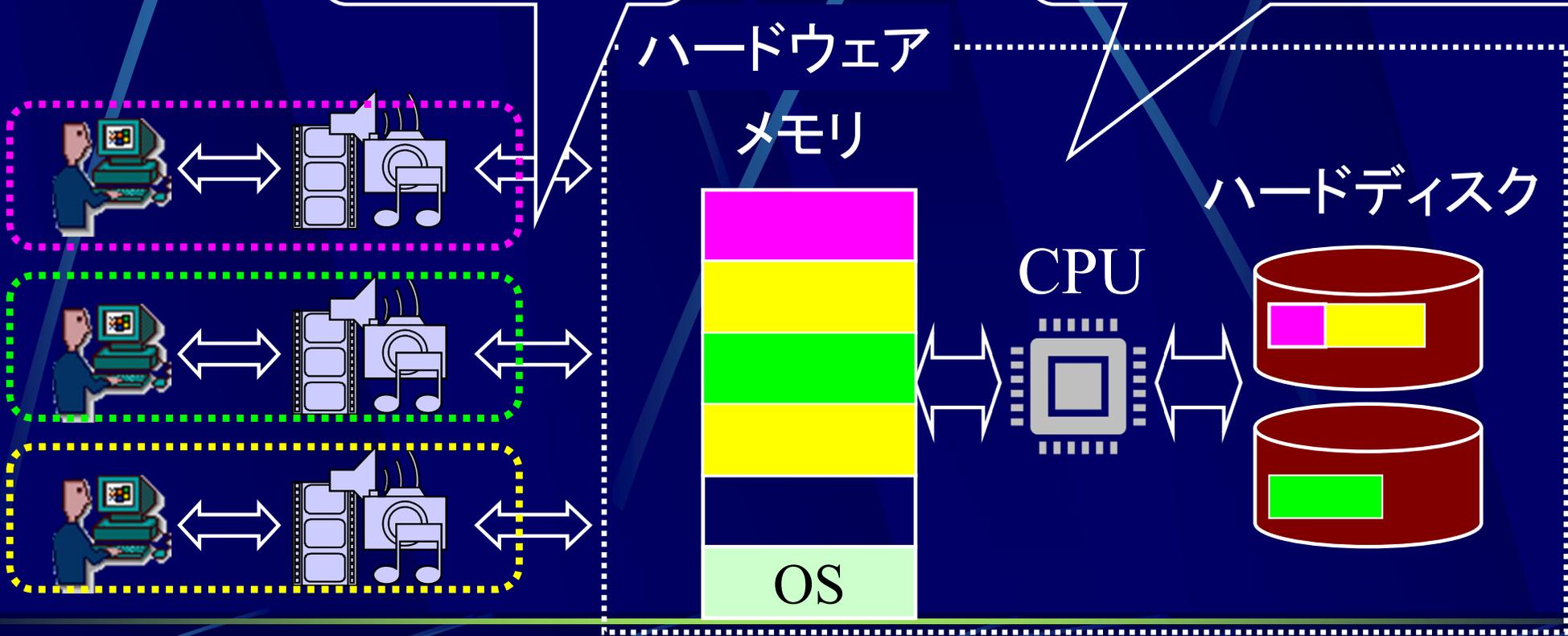
- アプリケーションプログラムの管理
- ハードウェア資源の管理
 - ハードウェア資源へのアクセスの制御
 - 性能の保証, 信頼性の保証



制御プログラム：性能の保証

応答時間の
保証をする

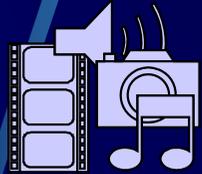
システム資源を
100%使用する



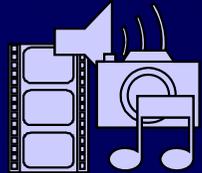
制御プログラム: 信頼性の保証

例: メモリ保護

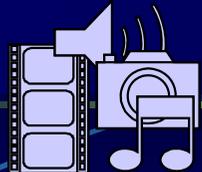
アプリケーション1



アプリケーション2



アプリケーション3



メモリ



1が2の領域へ
不当な書き込み

2が1の領域から
不当な読み込み

3がOSの領域に
不当な書き込み



アプリケーションを
停止する

OSの役割：仮想計算機(拡張計算機) (virtual machine, extended machine)

- ハードウェアを抽象化
- 抽象概念に対するアクセス手段の提供
- プログラミング環境の提供

仮想計算機： 計算機ハードウェアの抽象化



ファイル1

ファイル2

ユーザから見たデータ

使い易さを提供する

OS

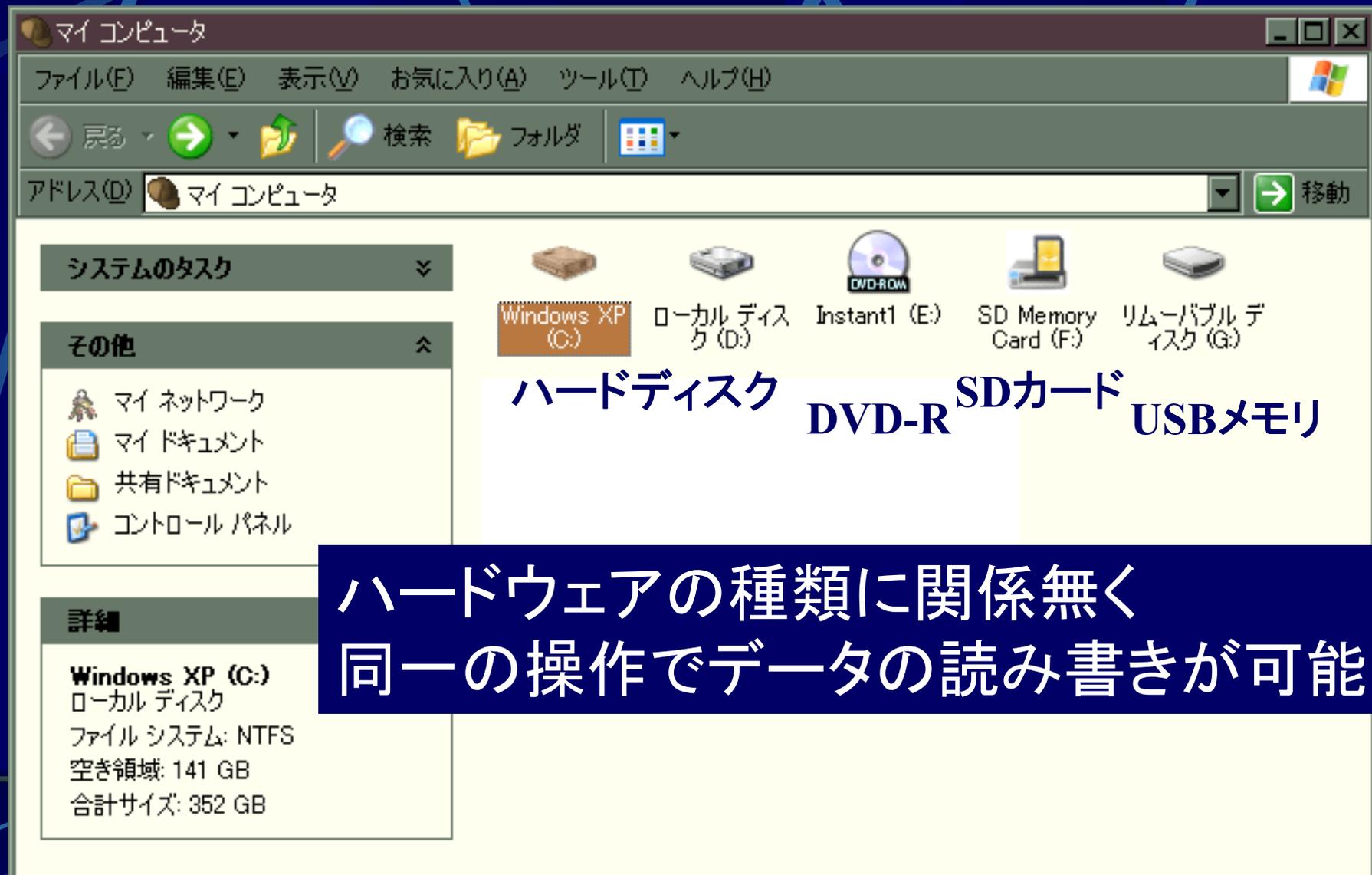
抽象化



アドレス	
0	
1	データ1
2	
3	未使用
4	
5	データ2
6	

実際のHDの中身

仮想計算機： 計算機ハードウェアの抽象化



仮想計算機： 計算機ハードウェアの抽象化

例：データのコピー



仮想計算機： 抽象概念に対するアクセス手段の提供

ハードウェア	抽象概念
メモリ	仮想アドレス空間
ディスク装置	ファイル
IO装置	
ネットワーク	プロセス
CPU	

ハードウェアに
直接アクセスさせない



不正アクセスの抑止
使い易さの提供

仮想計算機： 使い易さの提供

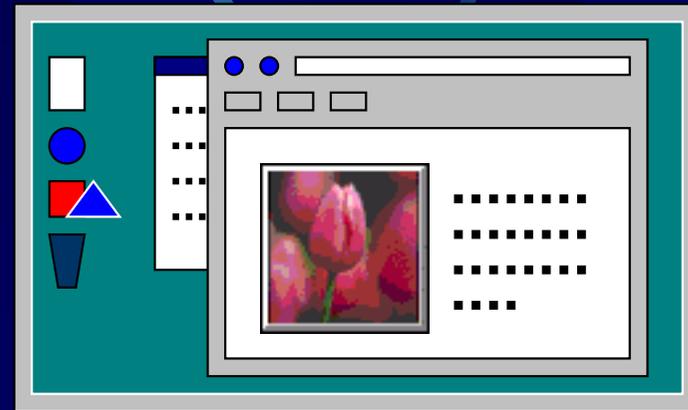
- ユーザがプログラムを簡単に実行できる



```
$ cd java  
$ ls  
hello.class hello.java  
lesson.pdf report.tex
```

A terminal window on a laptop screen displaying the output of a directory listing command.

コマンド入力により
プログラムを実行可能

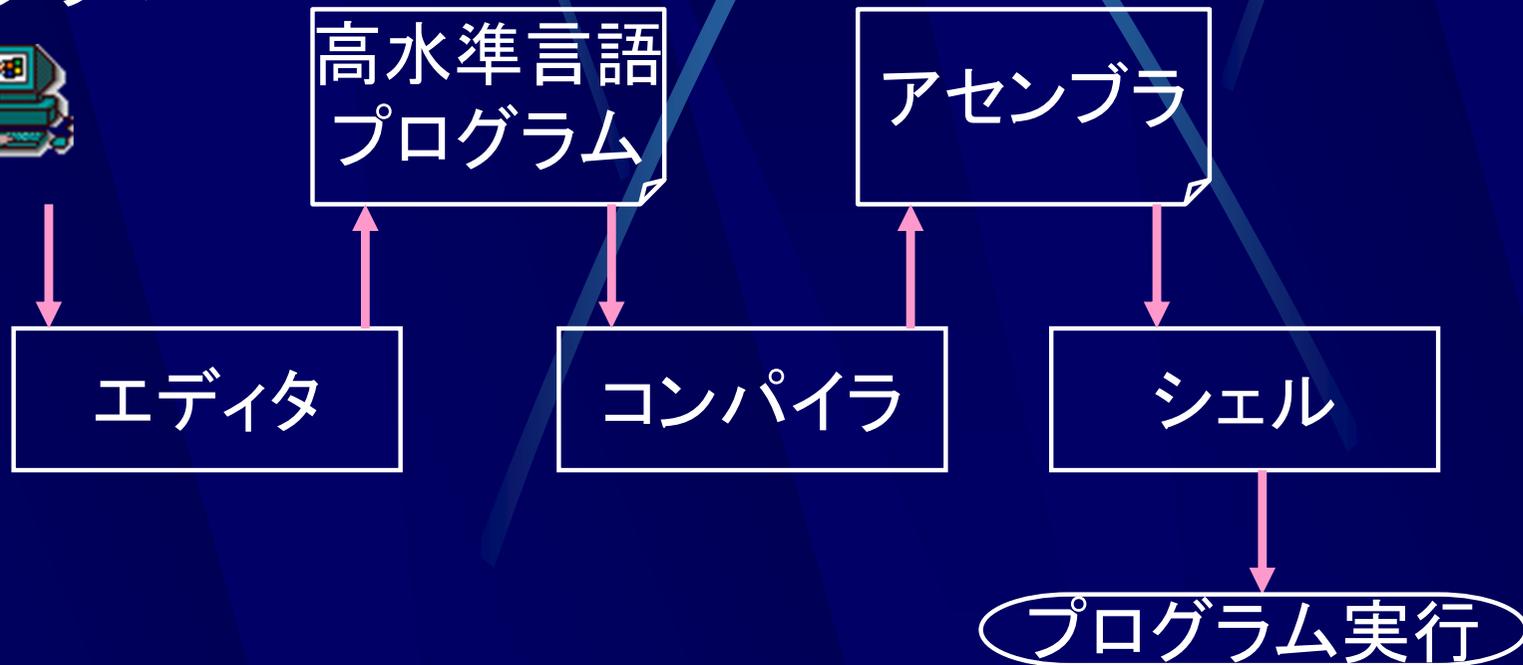


アイコンクリックにより
プログラムを実行可能

仮想計算機： プログラミング環境の提供

- コンパイラ, エディタ, シェル等と協調

プログラマ



もしOSが無かったら...

テキストエディタ

エディタ
機能

キーボード制御機能

ディスプレイ制御機能

磁気ディスク制御機能

ウェブブラウザ

ブラウズ
機能

キーボード制御機能

ディスプレイ制御機能

磁気ディスク制御機能

表計算

表計算
機能

キーボード制御機能

ディスプレイ制御機能

磁気ディスク制御機能

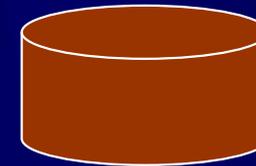
ハードウェア



キーボード



ディスプレイ



ハードディスク

アプリケーション毎に
各種ハード制御機能が必要

OSがあれば...

テキストエディタ

エディタ
機能

ウェブブラウザ

ブラウズ
機能

表計算

表計算
機能

OS

キーボード
制御機能

ディスプレイ
制御機能

磁気ディスク
制御機能

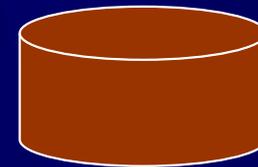
ハードウェア



キーボード



ディスプレイ



ハードディスク

各アプリケーションはOSとのみやり取りすればいい

OSの役割

- 資源管理者(resource manager)
 - ハードウェア資源,ソフトウェア資源の管理, ユーザへの割り当て
- 制御プログラム(control program)
 - アプリケーションプログラム, ハードウェアの管理
 - 性能の保証, 信頼性の保証
- 仮想計算機(virtual machine)
 - プログラミング環境を実権
 - “使い易さ”の提供

OSの歴史

第0世代	1940年代
第1世代	1950年代
第2世代	1960年代前半
第3世代	1960年代後半
第4世代	1980年代

OSの歴史(第0世代)

- 第0世代(1940年代)

- OSは存在しない
- ユーザは機械語でプログラムを作成,
コードの配線やパネルスイッチで入力

機械語

- 1と0のみで記述
- ハードに依存
(計算機ごとに異なる
プログラムが必要)

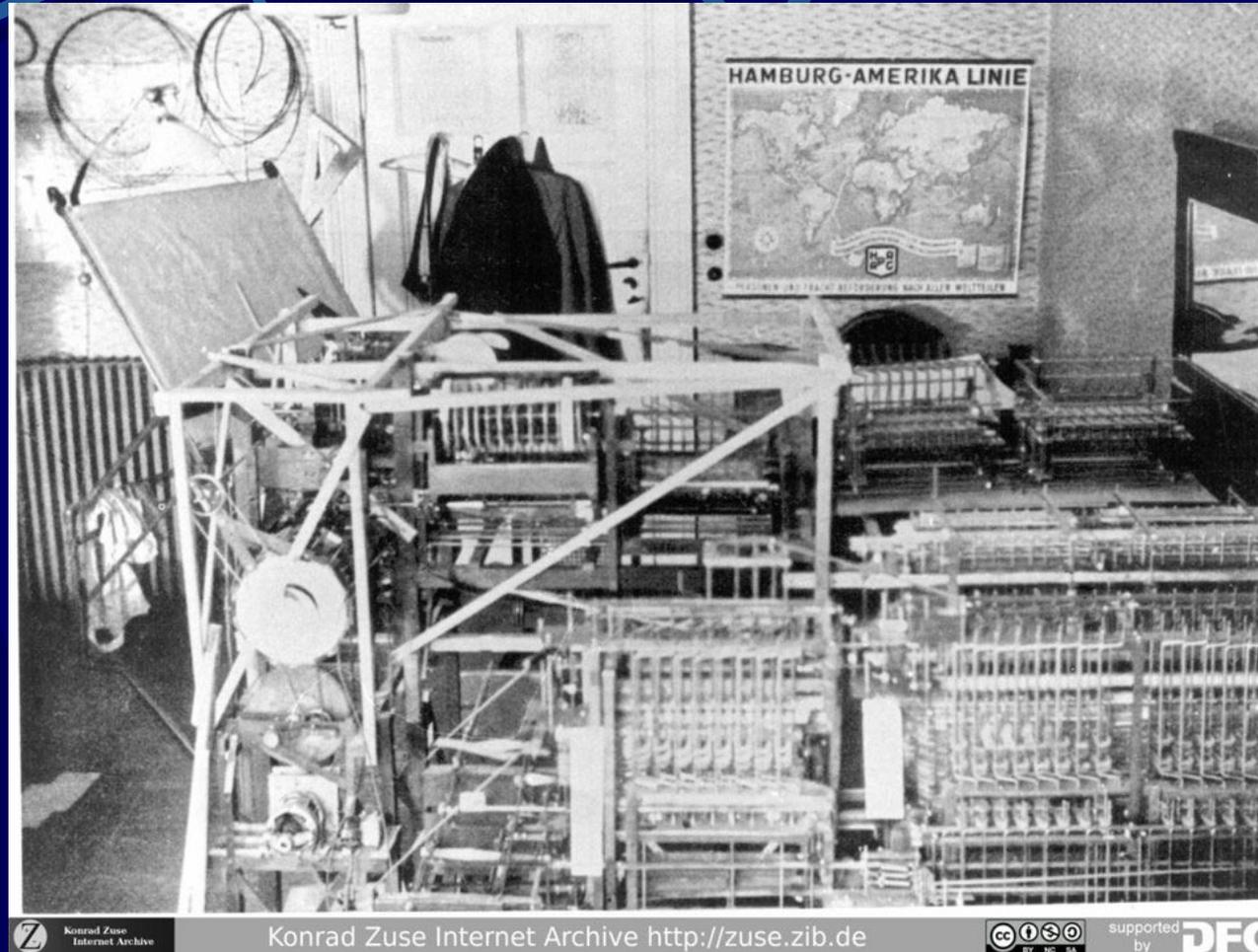
```
0101 B409
0102 BA09
0105 CD21
0107 CD20
:
```

Z1

● Z1

- 1938年 コンラッド・ツーゼ作
- 22ビットの機械式メモリ64個
- 22ビットの2進浮動小数点演算
- 世界最初のプログラム可能な計算機
 - ...の予定だったが誤作動が多く信頼性にかける
 - 使用済映画用フィルムでプログラム
- 1940年空爆により破壊される

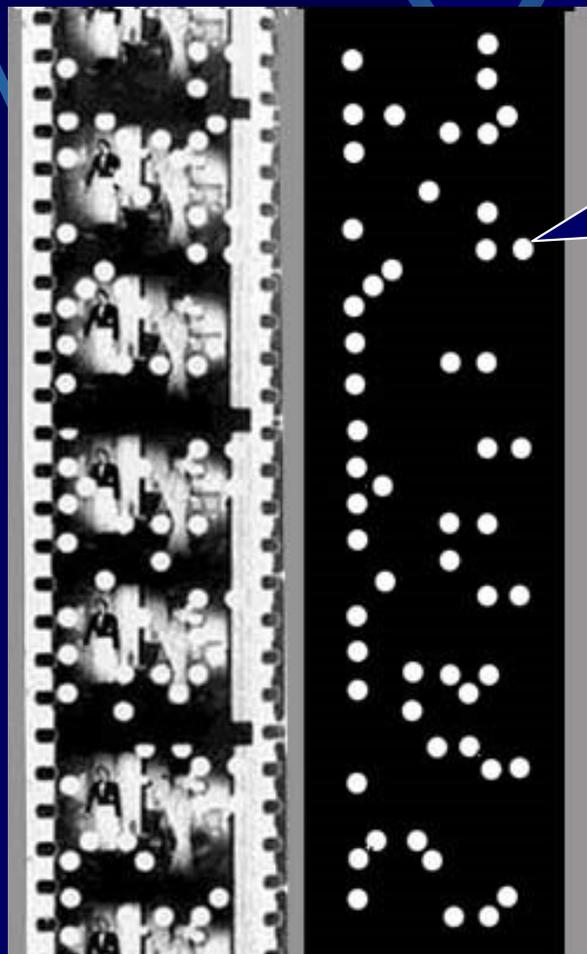
Z1



ツェーゼの計算機 Z1 [1]

[1] Konrad Zuse Internet Archive, <http://zuse.zib.de/>

Z1のプログラムフィルム



使用済フィルムに
穴を空けて
プログラム

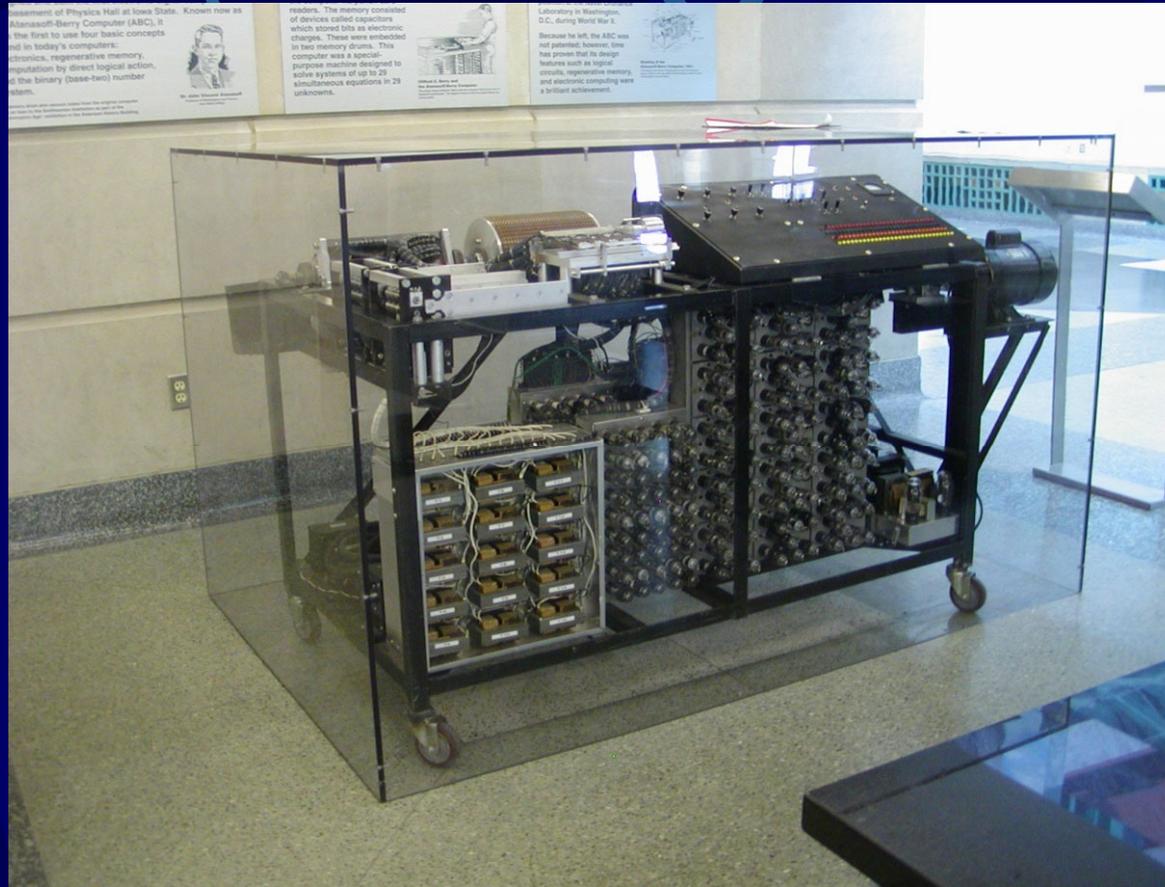
Z1のプログラムフィルム [1]

Atanasoff-Berry Computer

● ABC

- 1942年 ジョン・アタナソフとクリフォード・ベリー制作
- 50ビットの固定小数点演算
- 世界初の電子計算機
- 29元連立方程式を解ける
- ほぼ出来上がっていたが、アタナソフが戦時研究に駆り出されている間に解体された

Atanasoff-Berry Computer



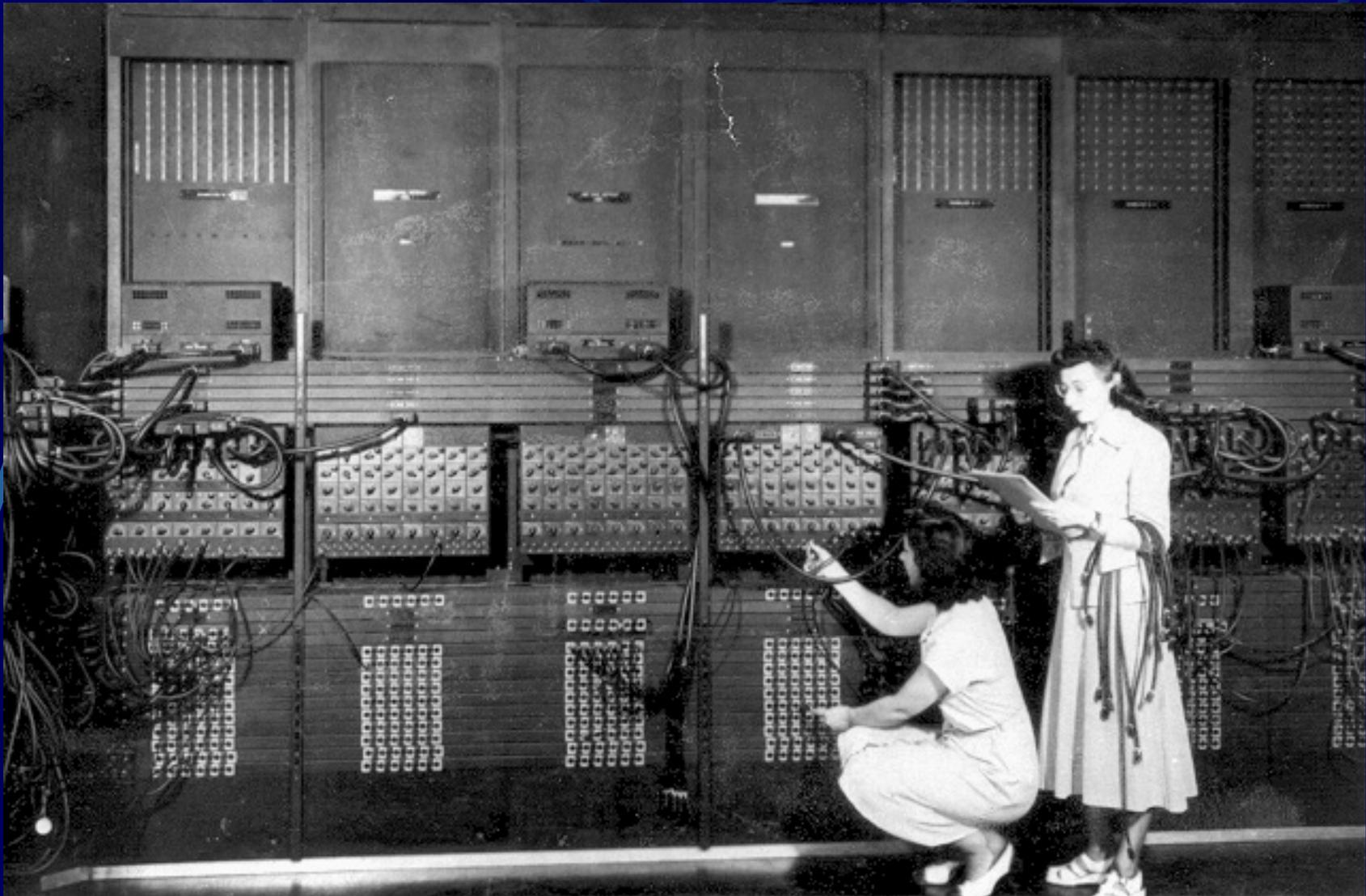
Atanasoff-Berry Computer (複製) [1]

[1] https://upload.wikimedia.org/wikipedia/commons/9/90/Atanasoff-Berry_Computer.jpg

Electronic Numerical Integrator and Computer

● ENIAC

- 1946年 ジョン・モークリーとジョン・エッカート制作
- 弾道表作製計算機
- 真空管式電子計算機
- プログラムは配線によりセット
 - 配線作業は1週間必要
- ~~「世界初の電子計算機」~~
 - 世界初は Atanasoff-Berry Computer



ENIACのプログラミング風景[1]

[1] John W. Mauchly and the Development of the ENIAC Computer,
An Exhibition in the Department of Special Collections,
Van Pelt Library, University of Pennsylvania,
<http://www.library.upenn.edu/exhibits/rbm/mauchly/jwminintro.html>

最初の計算機

年代	名称	備考	考案者・製作者
1938年 1941年	Z1 Z3	最初のプログラム可能な計算機	コンラッド・ツーゼ
1942年 未完成	ABC	最初の電子式計算機	ジョン・アタナソフ クリフォード・ベリー
1943年	コロッサス	最初の真空管式大規模機械	アラン・チューリング トム・フラワーズ
1944年	ASCC	最初に実用化されたプログラム可能な計算機	ハワード・エイケン
1946年	ENIAC	最初の実用化された電子式計算機	ジョン・モークリー ジョン・エッカート
1948年	SSEM	最初に稼働したフォン・ノイマン型計算機	フレデリック・ウィリアムズ トム・キルバーン
1949年	EDSAC	最初に実用化されたフォン・ノイマン型計算機	モーリス・ウィルクス
1951年	EDVAC	最初にフォン・ノイマン型を提唱した計算機	ジョン・モークリー ジョン・エッカート

OSの歴史(第1世代)

- 第1世代(1950年代)
 - バッチ処理(batch processing)
 - 複数のプログラムを連続して処理

同様なジョブを一括処理(batch)することによりセットアップ時間を軽減

バッチ処理

プログラム1



プログラム2



プログラム3

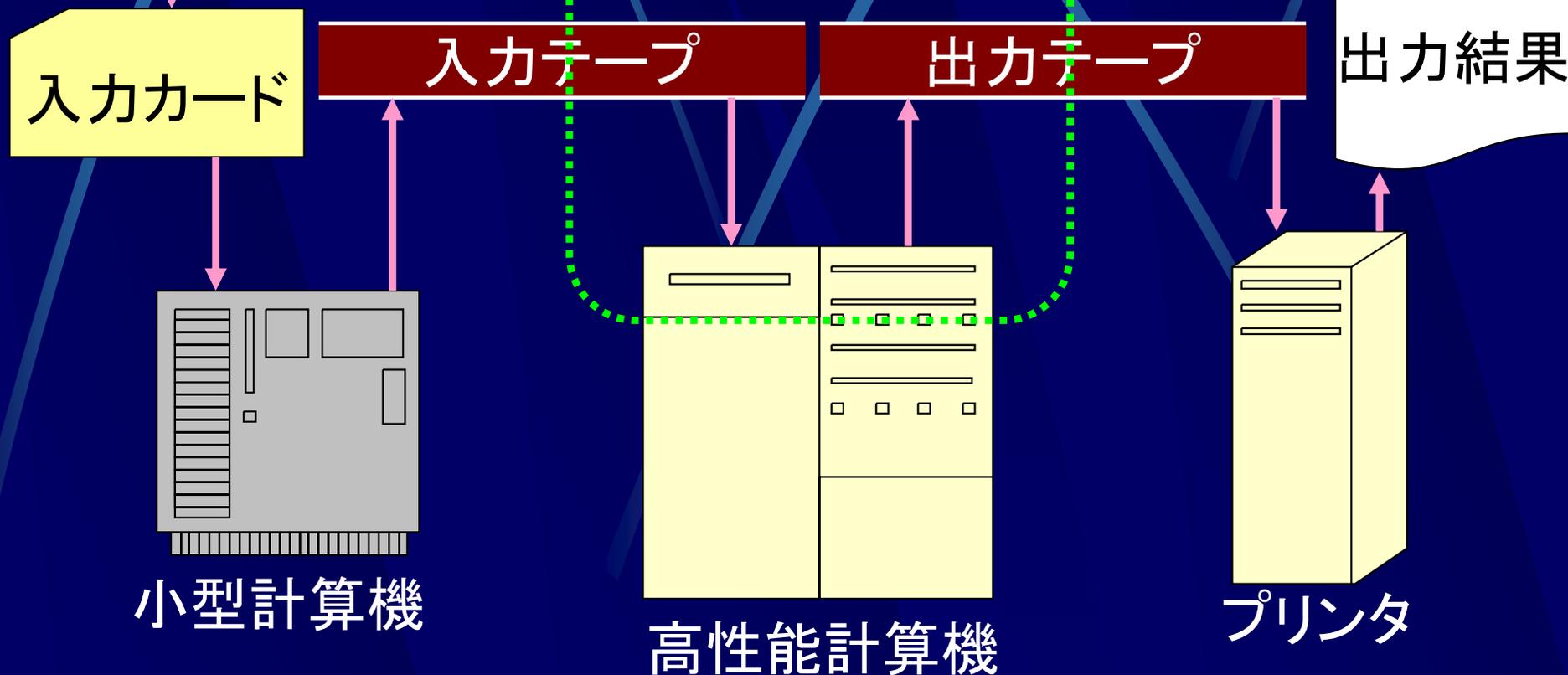


プログラム4

OSの歴史(第1世代): バッチ処理の流れ

この部分の処理時間を短縮する

プログラマ



遅い部分は衛星計算機(satellite computer)で別処理

OSの歴史(第2世代)

- 第2世代(1960年代前半)
 - マルチプログラミング(multiprogramming)
 - 並行処理(concurrent processing)
 - 複数のプログラムを切り替えて実行
 - タイムシェアリング(time-sharing)
 - 複数のユーザの仕事进行处理
 - 仮想記憶(virtual address)
 - 主記憶よりも大きい記憶空間の提供

OSの歴史(第2世代): マルチプログラミング (multiprogramming)

- 複数のプログラムを同時にメモリに置く
メモリ

プログラム1

プログラム2

プログラム3

OSの歴史(第2世代): マルチプログラミング

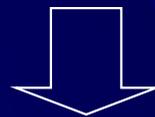
- プログラムにはユーザの入力やIO処理が含まれる



ユーザからの入力待ち

モニタからの返答待ち

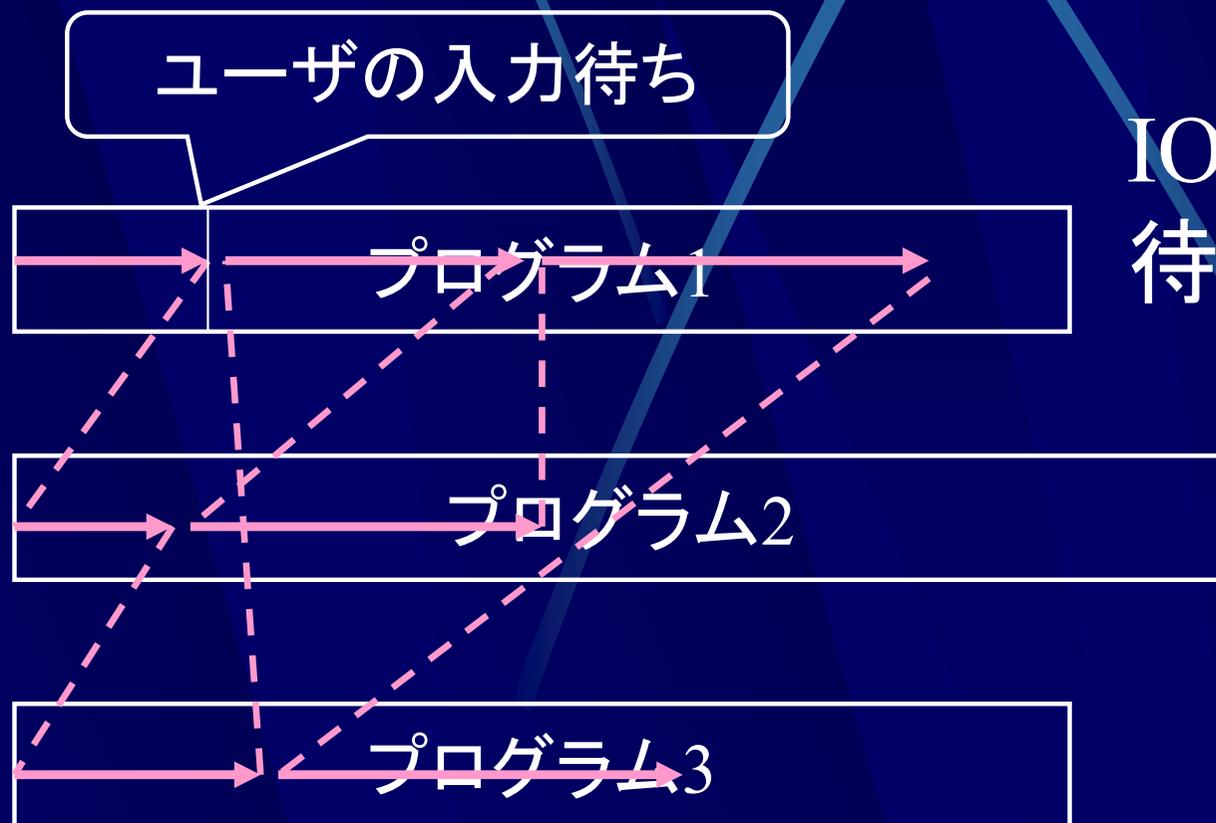
待っている間はプロセッサはアイドル(何もしない)状態



待ち時間の間に他のプログラムを行える

OSの歴史(第2世代): マルチプログラミング

- CPUはプログラムを高速に切り替えて実行



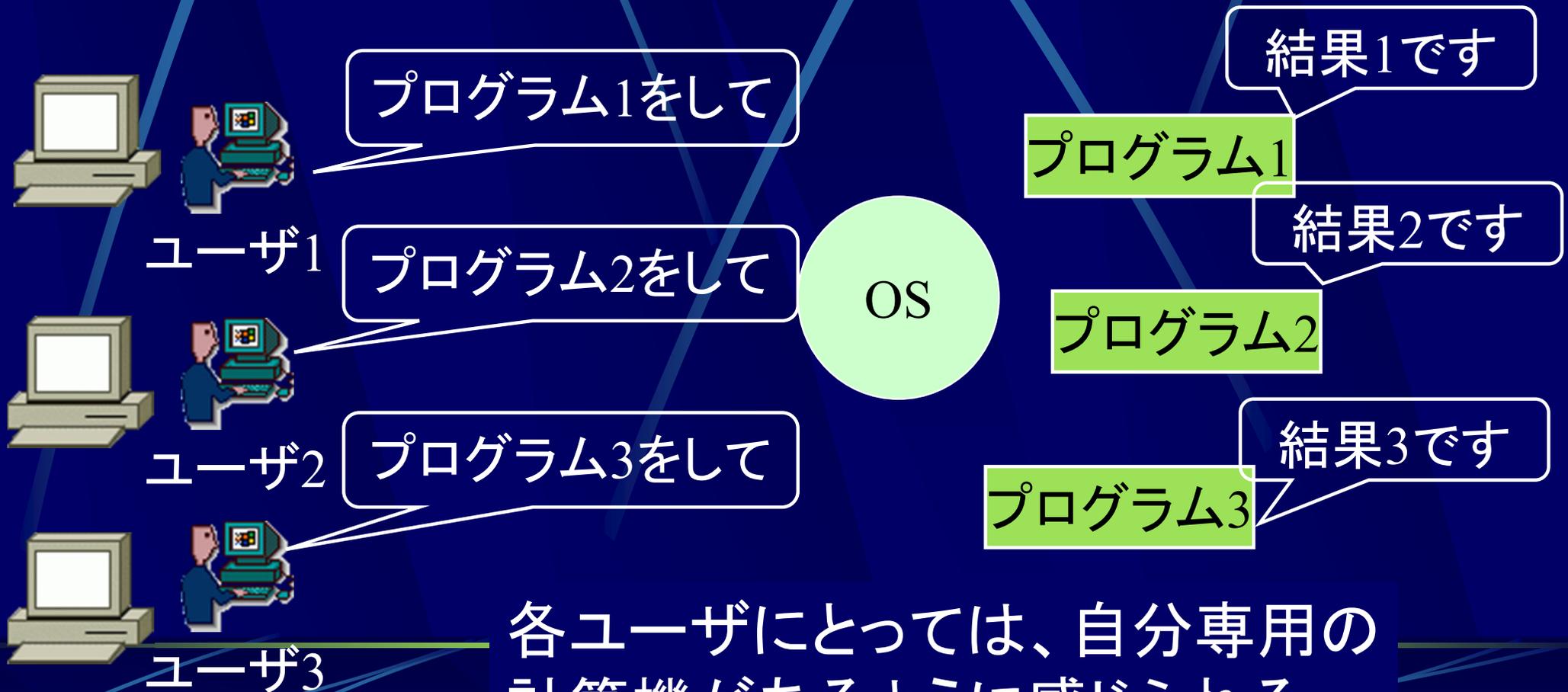
IO等の無駄な
待ち時間を回避

OSの歴史(第2世代): マルチプログラミング

- マルチプログラミングに必要な能力
 - システムに備えられたIOルーチン
 - メモリ管理
 - 複数のプログラムをメモリに割り当てる
 - CPUスケジューリング
 - 複数のプログラムからどのプログラムを次に実行するか決定する
 - デバイスの割り当て

OSの歴史(第2世代): タイムシェアリング (time-sharing)

- 複数のユーザの仕事に対話的に処理

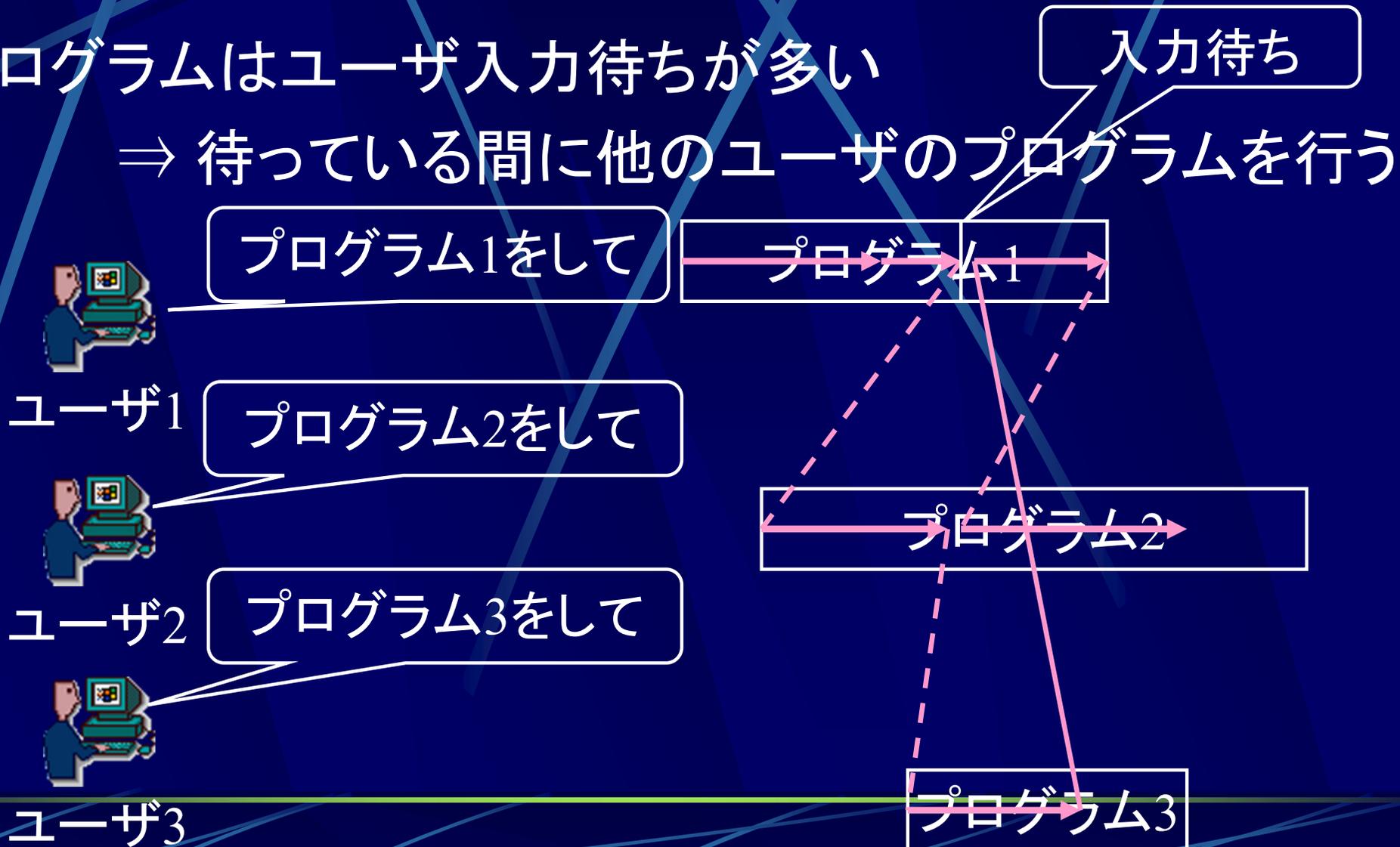


各ユーザにとっては、自分専用の
計算機があるように感じられる

OSの歴史(第2世代): タイムシェアリング

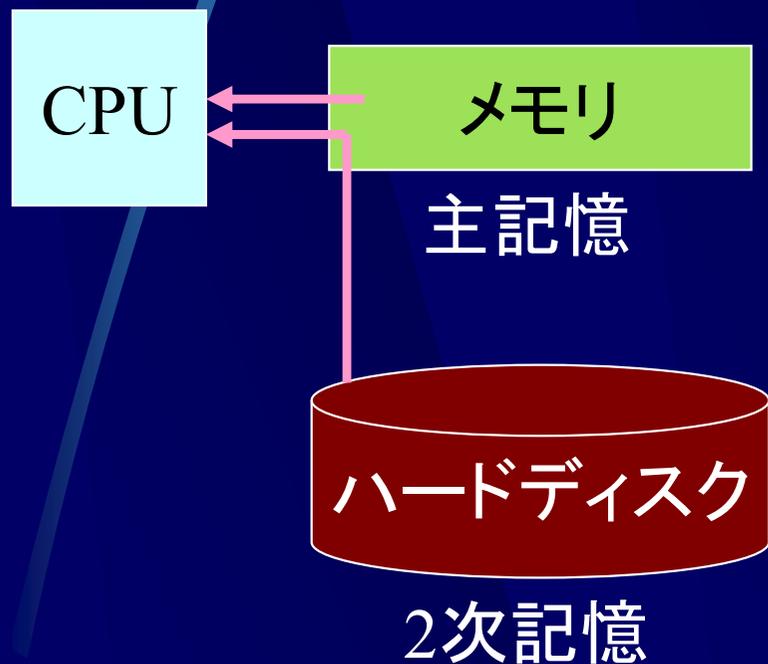
プログラムはユーザ入力待ちが多い

⇒ 待っている間に他のユーザのプログラムを行う



OSの歴史(第2世代): 仮想記憶 (virtual address)

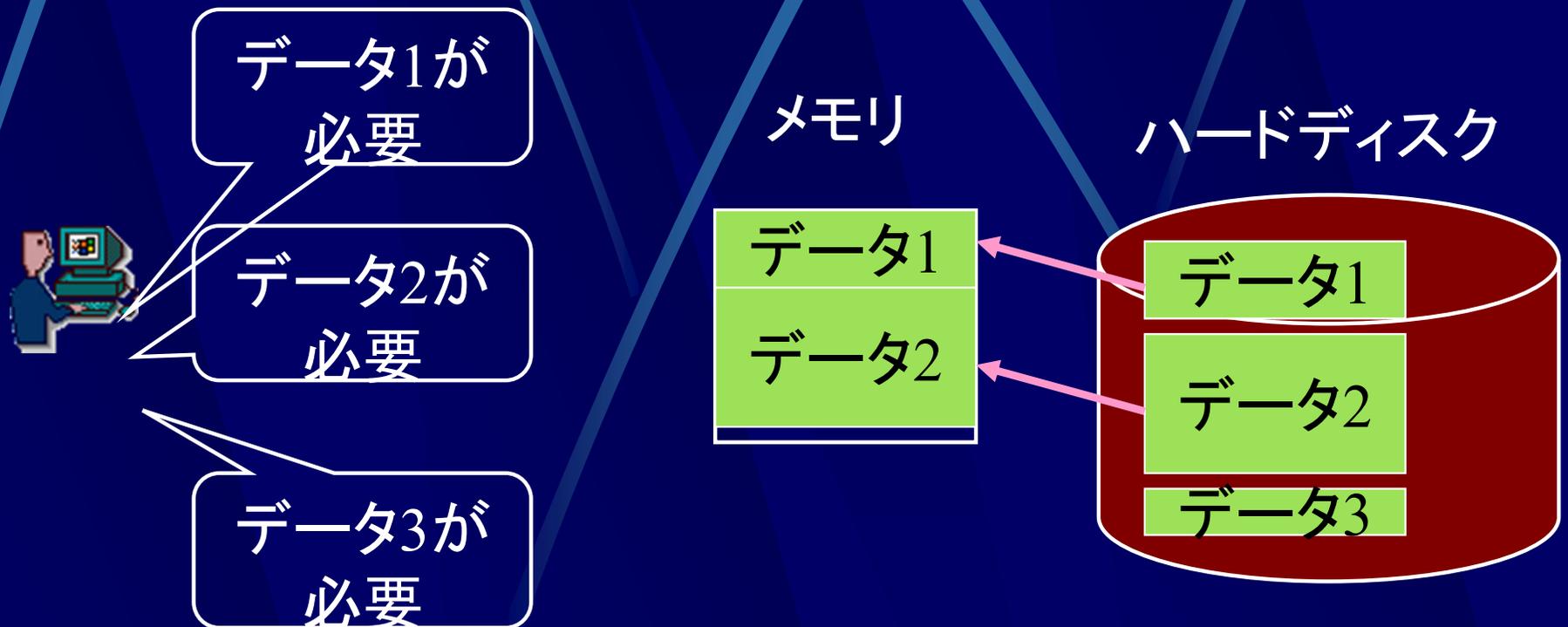
- 主記憶より大きい記憶空間の提供



- CPUが直接読み書き可能
- 高速
- 容量が小さい
- 一旦データをメモリに移さないとCPUが読み書きできない
- 低速
- 容量が大きい

ハードディスクを仮想的なメモリとして使用する

OSの歴史(第2世代): 仮想記憶

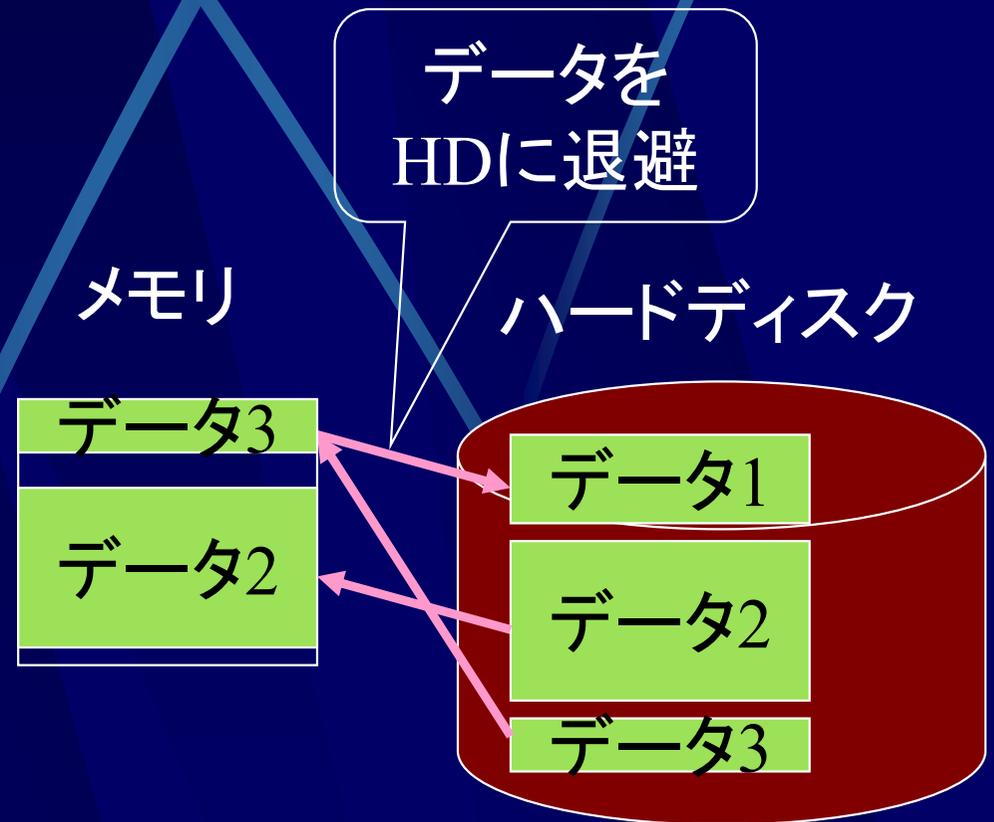


必要なデータを適時メモリに読む込む

OSの歴史(第2世代): 仮想記憶



データ3が
必要



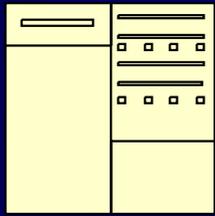
メモリに入りきらない場合は
データの入れ替え(スワップ)をする

OSの歴史(第3世代)

- 第3世代(1960年代後半)
 - 汎用大型計算機
 - IBM System/360 の OS/360
 - MULTICS
 - シミュレータ, エミュレータ

OSの歴史(第3世代): 専用機から汎用機へ

第2世代初期の計算機

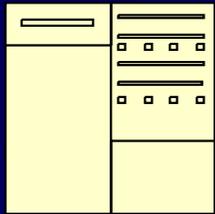


計算機1

+

事務用ソフトウェア

= 事務専用機

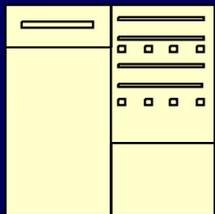


計算機2

+

計算用ソフトウェア

= 科学計算専用機



計算機3

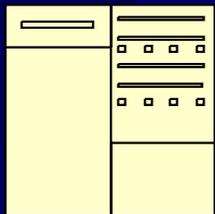
+

通信用ソフトウェア

= 通信専用機

OSの歴史(第3世代): 専用機から汎用機へ

第2世代後期の計算機



計算機

+

事務用ソフトウェア

計算用ソフトウェア

通信用ソフトウェア

1台の計算機を
様々な目的に
使用できる

OSの歴史(第3世代): 専用機から汎用機へ

- 多くのユーザが汎用性を求めた
 - 科学技術計算
 - 商用計算
 - 事務処理
 - 通信

多数の機能を持つ反面、
システムが巨大化してしまった

汎用大型計算機

OSの歴史(第3世代): IBM System/360

- 1964年IBMが開発
- 初の本格的OS OS/360 を搭載



System/360[1]

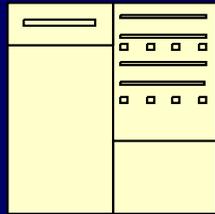
[1] System 360 From Computers to Computer Systems, IBM

<http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/system360/impacts/>

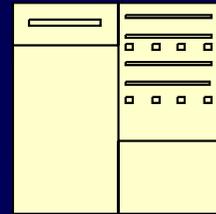
OSの歴史(第3世代): IBM System/360

- IBMは System/360 の後継機を次々と発表
 - System/370, 4300, 3080, 3090

計算機ファミリー



計算機1

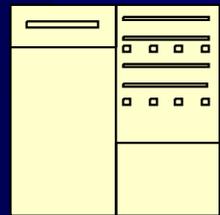


1の後継機

後継機は計算機1のソフトウェアをそのまま使える

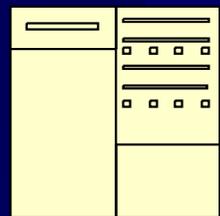
OSの歴史(第3世代)

● 第2世代以前の計算機



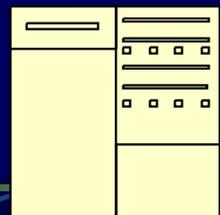
計算機1

計算機1用の
ソフトウェア



計算機2

計算機2用の
ソフトウェア



計算機3

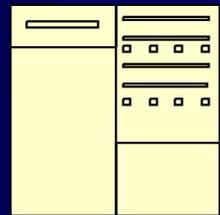
計算機3用の
ソフトウェア

計算機間で
ソフトウェアの
互換性無し

計算機ごとに
ソフトウェアの
開発が必要

OSの歴史(第3世代)

● 第3世代以降の計算機



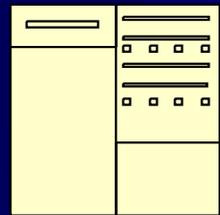
+

OS



汎用ソフトウェア

計算機1



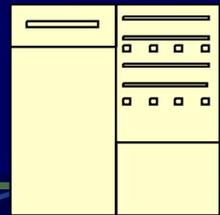
+

OS



汎用ソフトウェア

計算機2



+

OS



汎用ソフトウェア

計算機3

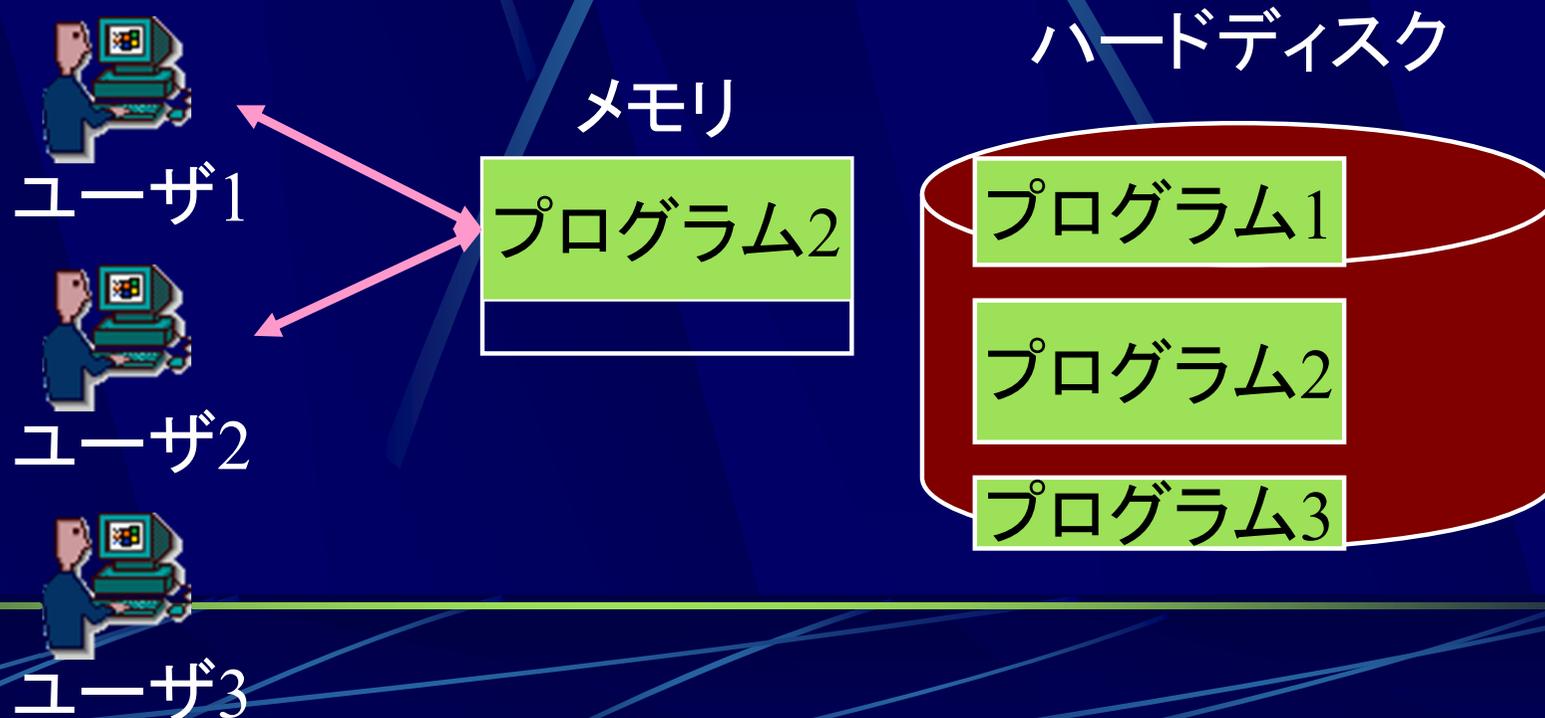
異なる計算機に
共通の
ソフトウェアを
使用できる

OSの歴史(第3世代):

CTSS(the compatible time-sharing system)

- 1963年MITが開発

- 複数のユーザのプログラムを切り替えながら実行可能



OSの歴史(第3世代): MUTICS

(multiplexed information and computing service)

- 1965年MITとベル研が共同開発
 - CTSSの流れを汲むOSの原型
 - 仮想記憶, プロセスの概念, プロテクションの概念等の重要なアイデアが盛り込まれる

しかし実用化はせず

1965年当時

- ハードウェア能力が無い
- ソフトウェア技術が無い

OSの歴史(第3世代): UNICS

- 1968年ベル研が開発
 - MULTICSの失敗(複雑過ぎ, 大き過ぎ)を踏まえたシンプルで小さいOS

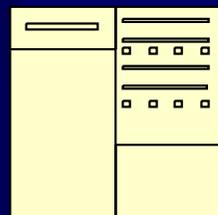
後に UNIX と改名

ただし現在は巨大化

OSの歴史(第3世代): シミュレータ, エミュレータ (simulator, emulator)

- 他の計算機の命令をシミュレート可能

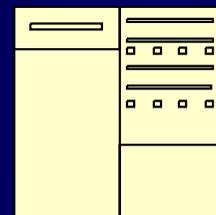
計算機1用プログラム



計算機1

計算機1用プログラム

計算機1のシミュレータ



計算機2

異なる計算機で同一のソフトウェアが使用可能

OSの歴史(第3世代): シミュレータ, エミュレータ

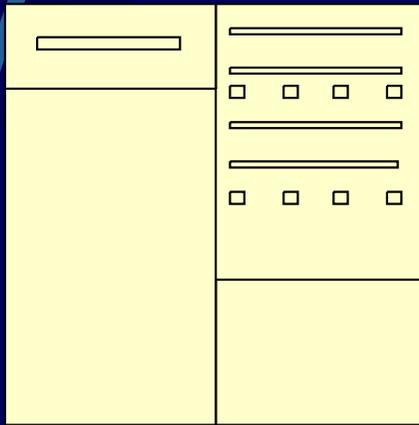
	シミュレータ	エミュレータ
構成	ソフトウェア	マイクロプログラム 機械語
実行速度	遅い	速い
費用	安い	高い
開発難易度	低い	高い

OSの歴史(第4世代)

- 第4世代(1970年代後半)
 - 計算機ネットワーク
 - オンライン処理
 - データベース
 - パーソナルコンピュータ
 - ワークステーション

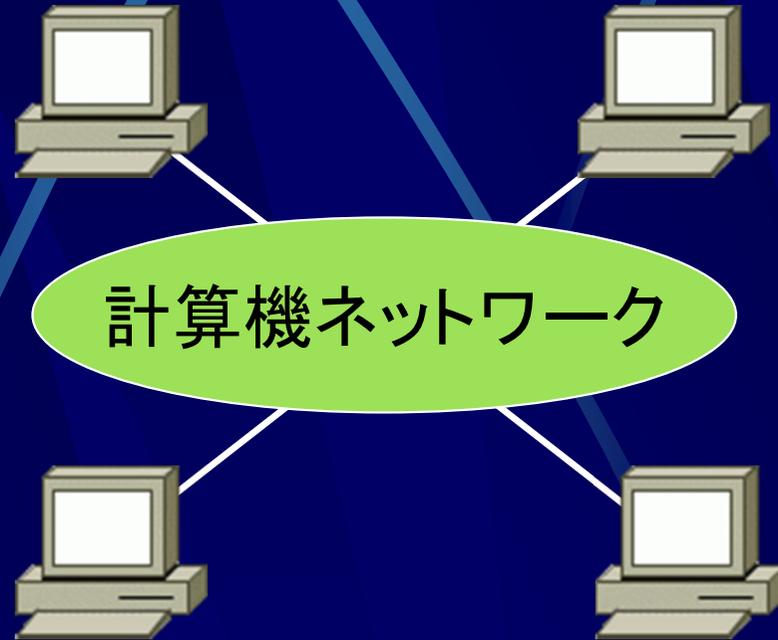
OSの歴史(第4世代): 計算機ネットワーク

第3世代以前の計算機



スタンドアロンで使用

第4世代以降の計算機



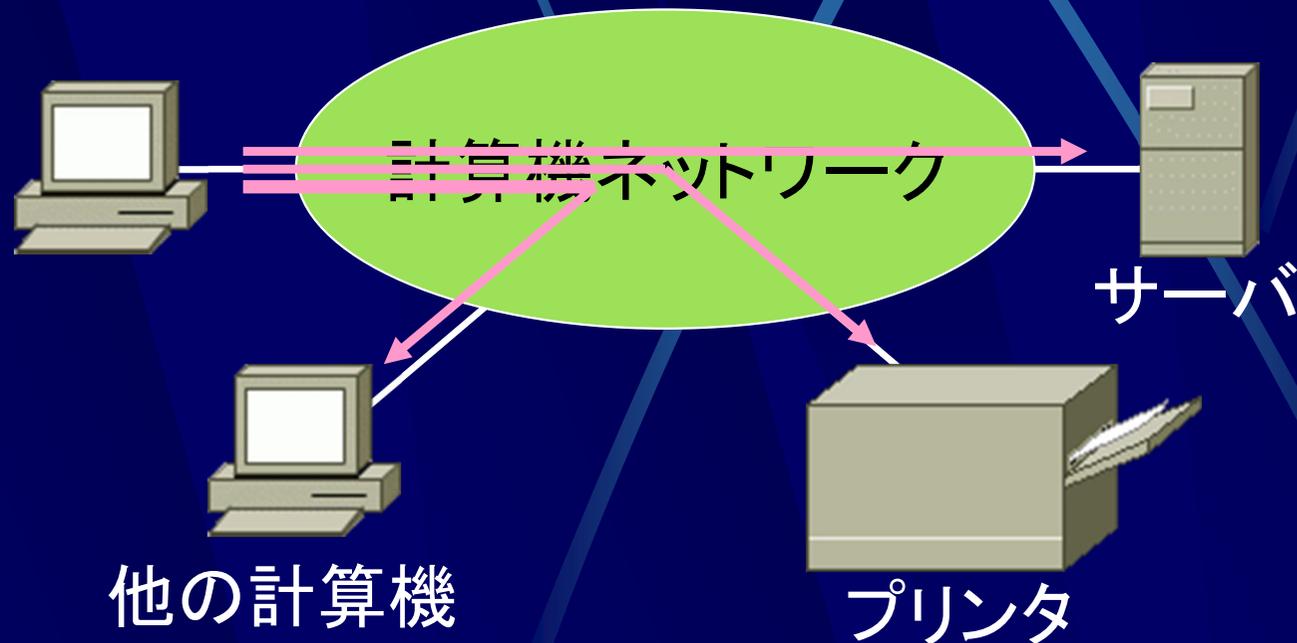
計算機ネットワークで接続して
互いに通信を行う

LAN(local area network)が誕生

OSの歴史(第4世代): 計算機ネットワーク

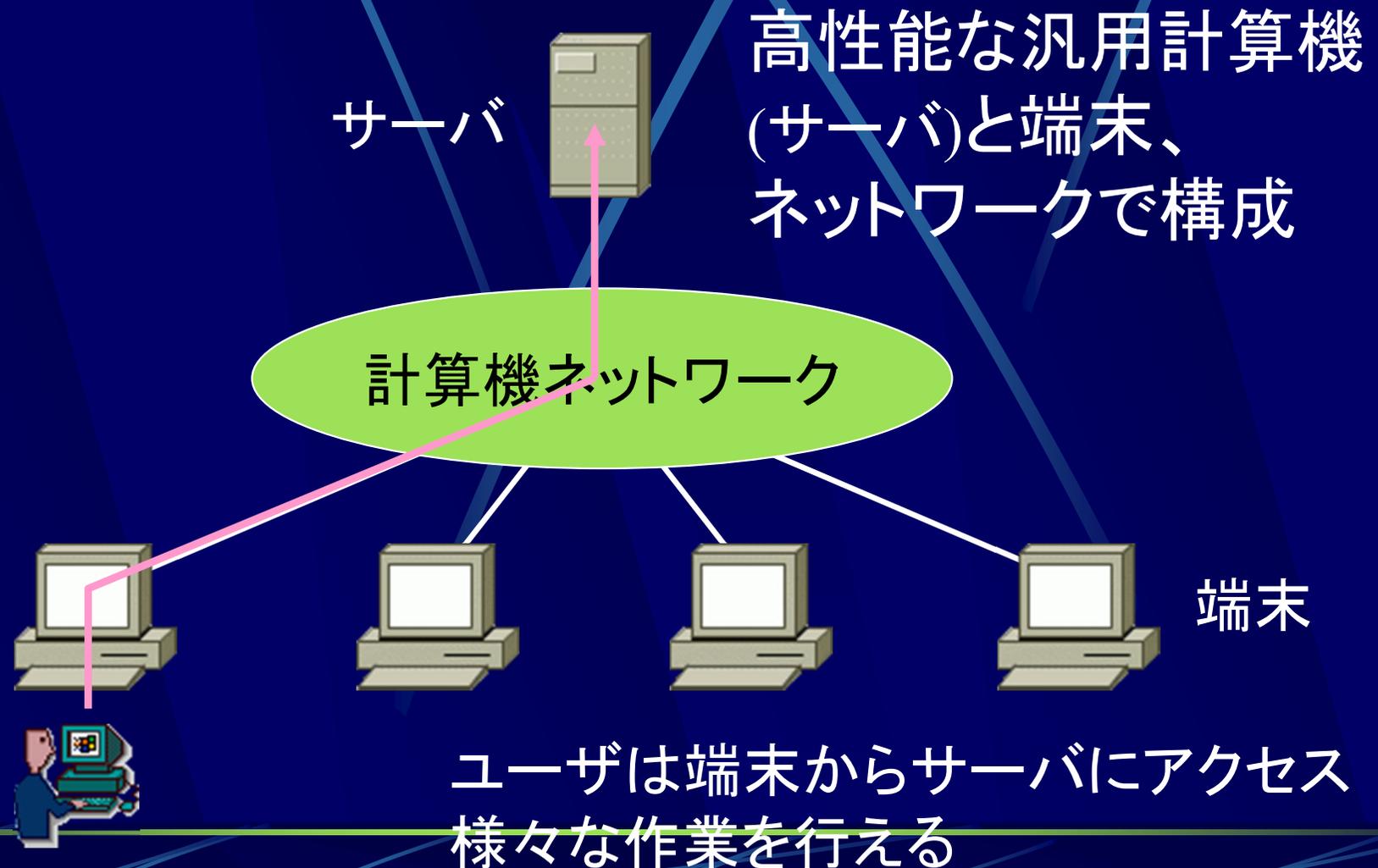
1970年代初期	ハワイ大学で研究され、「Alohanet」が開発 ハワイ諸島の端末が共有無線周波数帯へ構造 化アクセス
1972年	米Xerox社のPARCが自社PC(Alto)用に「Alto Aloha Network」
1973年	Alto以外のコンピュータにも対応させ、 「Ethernet」と命名
1980年	米DEC社、米Intel社、米Xerox社の3社連合 (DIX)によって、最初のイーサネット規格が発表 される

OSの歴史(第4世代): オンライン処理



計算機ネットワークを通して
様々な処理を行うことができる

OSの歴史(第4世代): ワークステーション



OSの歴史(第4世代): パーソナルコンピュータ

- 1975年 MITS がAltair 8800 開発
- 1977年 Apple computer がApple II 開発
- 1981年 IBMが IBM PC 5150 開発

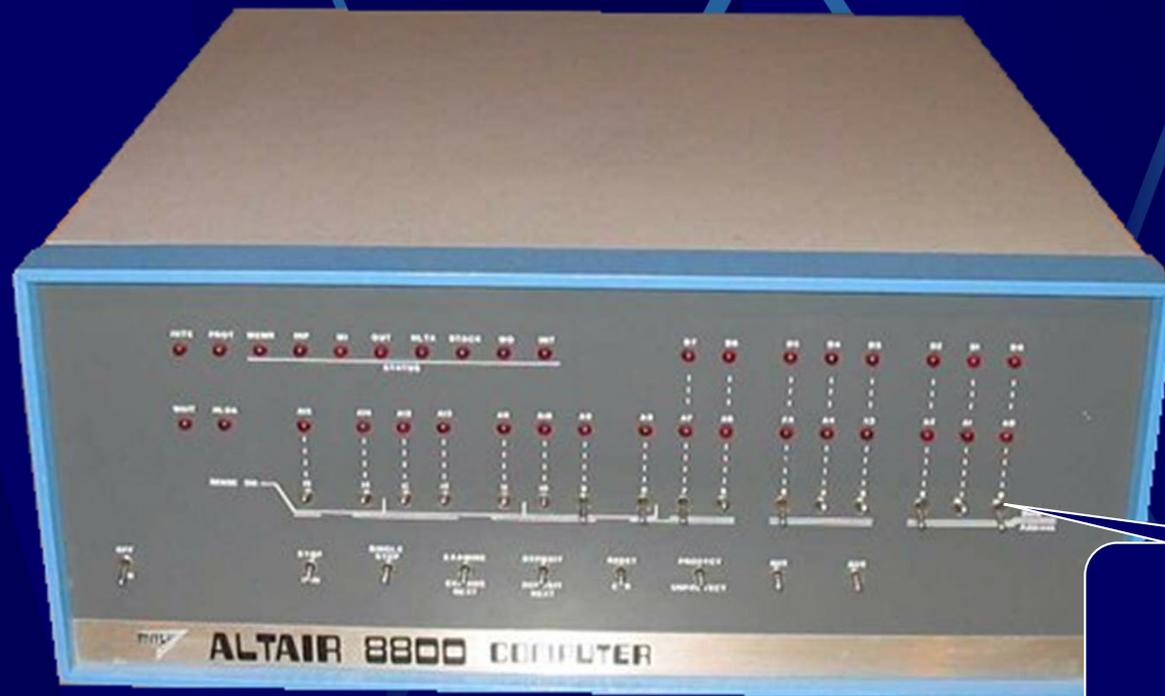
以降様々なパーソナルコンピュータが開発される

組織に1台の時代から、個人に1台の時代へ

Altair 8800

- Altair 8800
 - 1974年 Micro Instrumentation and Telemetry Systems が開発
 - コンピュータキット
 - 組み立てキットで販売
 - 機体前面のスイッチでプログラム
 - CPU : Intel 8080 2MHz
 - メモリ : 256 B

Altair 8800



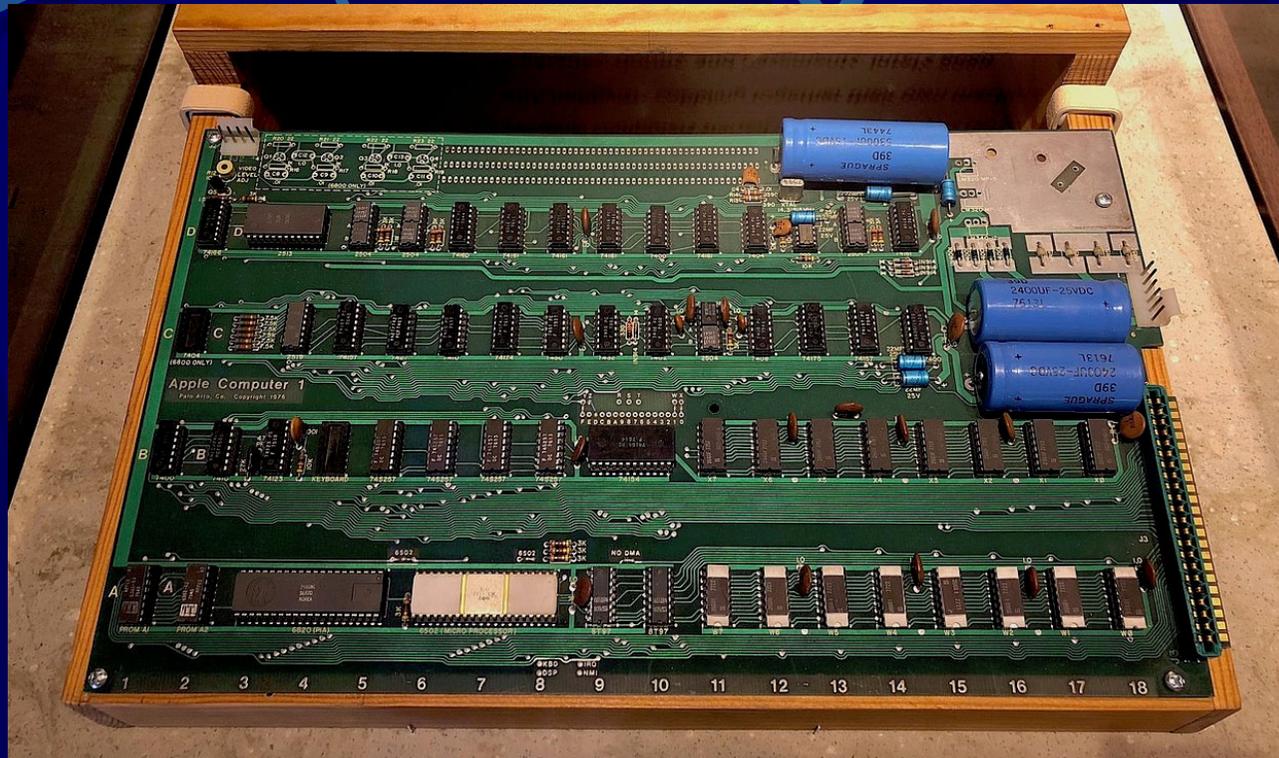
スイッチで
プログラム

Altair 8800 [1]

Apple I

- Apple I
 - 1976年 Apple が開発
 - ワンボードマイクロコンピュータ
 - キーボード等は別途必要
 - CPU : DOS 6502 1 MHz
 - メモリ : 4 KB

Apple I



Apple I [1]

基盤のみ モニタ・キーボードは別途必要

[1] https://upload.wikimedia.org/wikipedia/commons/1/10/Apple_1_Woz_1976_at_CHM.agr_cropped.jpg

Apple II

- Apple II
 - 1977年 Apple が開発
 - パーソナルコンピュータ
 - キーボードが本体に組み込まれる
 - OS : Apple DOS
 - CPU : MOS Technology 6502 1.023 MHz
 - メモリ : 4 KB

Apple II



本体に
キーボード

Apple II [1]

[1] https://upload.wikimedia.org/wikipedia/commons/9/98/Apple_II_typical_configuration_1977.png

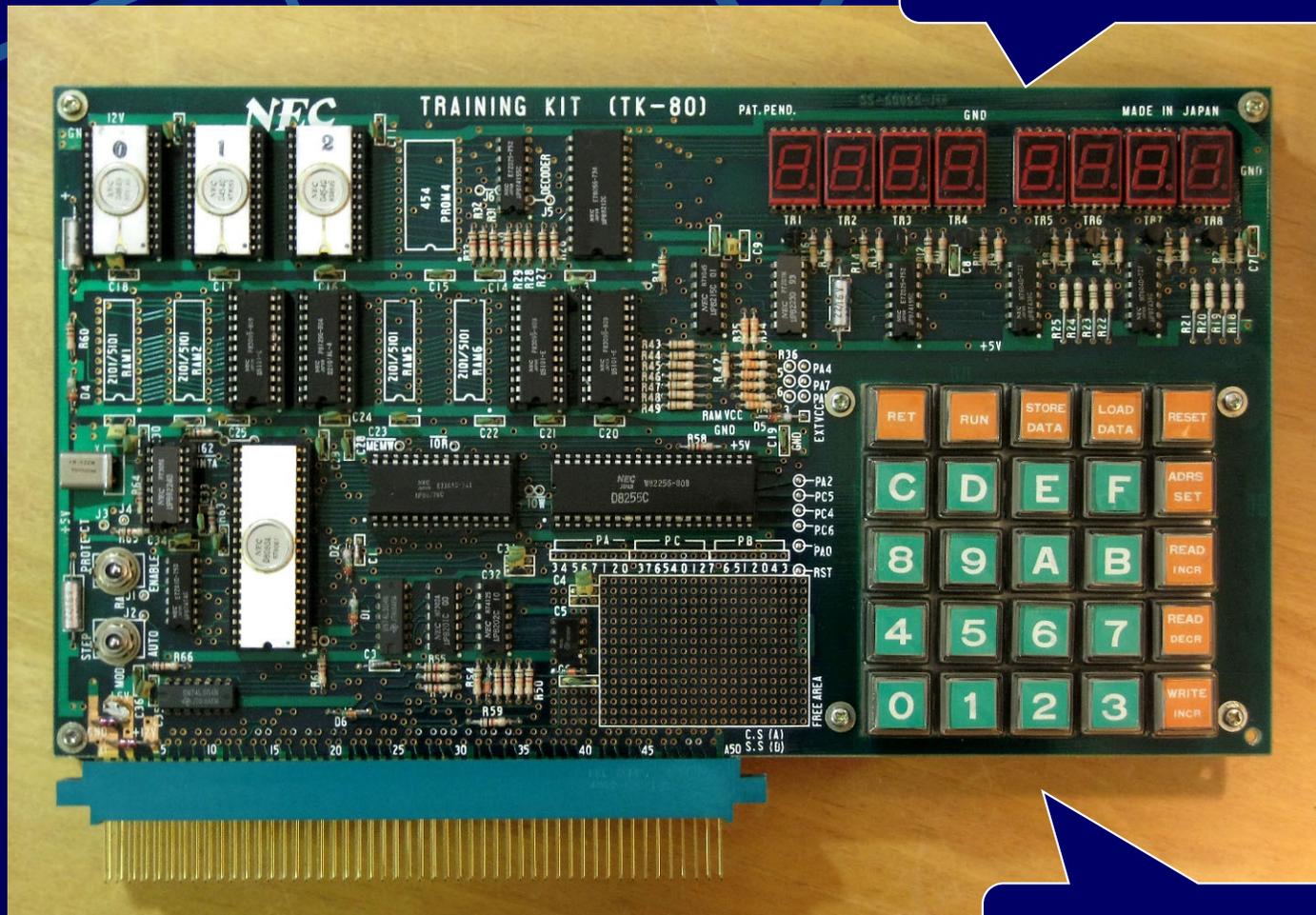
TK-80

● TK-80

- 1976年 日本電気 (NEC) が開発
- ワンボードマイクロコンピュータ
 - ボード単体で使用可能
 - 16進キーボードでプログラム
 - 7セグメントLED で値表示
- CPU : NEC μ PD8080A 2.048 MHz
- メモリ : 512 B

TK-80

7セグメントLED



16進キーボード

TK-80 [1]

PC-8001

- PC-8001

- 1979年 日本電気 が開発
- パーソナルコンピュータ
 - キーボードが本体に組み込まれる
- OS : N-BASIC
- CPU : NEC μ PD780C-1 4 MHz
- メモリ : 16 KB

PC-8001



本体が
キーボードと
一体化

PC-8001 [1]

PC-9801

- PC-9801

- 1982年 日本電気が開発
- パーソナルコンピュータ
- 一時は日本国内で圧倒的シェアを占める
- OS : MS-DOS
- CPU : NEC μ PD8086 5 MHz
- メモリ : 128 KB

PC-9801



PC-9801 [1]

[1] <https://upload.wikimedia.org/wikipedia/commons/4/4f/PC-9801-1st-001.jpg>

IBM PC 5150

- IBM personal computer model 5150
 - 1981年 IBM が開発
 - ビジネス用パーソナルコンピュータ
 - OS : PC DOS
 - CPU : Intel 8088 4.77 MHz
 - メモリ : 16~256 KB

IBM PC 5150



Macintosh 128K

- Macintosh 128K
 - 1984年 Apple が開発
 - 一体型パーソナルコンピュータ
 - グラフィカルユーザインタフェース
 - マウスによる操作
 - OS : System 1.0
 - CPU : MC68000 7.8336 MHz
 - メモリ : 128 KB

Macintosh 128K



マウスで
操作可能

Macintosh 128K [1]

[1] https://upload.wikimedia.org/wikipedia/commons/c/cf/Macintosh%2C_Google_NY_office_computer_museum_cropped.png

パーソナルコンピュータの歴史

発売年	機種	販売元	備考
1975	Altair 8800	MITS	組み立てキットで販売 スイッチでプログラム
1976	Apple I	Apple	基盤のみ
1976	TK-80	日本電気	基盤に16進キーボードと7セグメントLED 追加機器無しで使用可能
1977	Apple II	Apple	キーボードが本体に組み込み
1979	PC-8001	日本電気	日本のパソコンの先駆け
1981	IMB PC 5150	IBM	ビジネス用パソコン
1982	PC-9801	日本電気	一時は圧倒的国内シェアを占める
1984	Macintosh 128K	Apple	GUI採用 マウスで操作可能

計算機の世代

世代	素子	プログラミング・言語	特徴	使用者
第0世代 1940年代	機械式	ケーブルの差し入れ・機械語	黎明期 OSは存在しない	開発者
第1世代 1950年代	真空管	ケーブルの差し入れ・機械語	バッチ処理	開発者
第2世代 1960年代 前半	トランジスタ	コマンドベース インタフェース・ 低水準言語	マルチプログラミング タイムシェアリング 仮想記憶 専用ソフトウェア	専門技術者
第3世代 1960年代 後半	IC, LSI	CUI+ 階層メニュー・ 高水準言語	汎用大型計算機 シミュレータ, エミュレータ 汎用ソフトウェア	専門的職人
第4世代 1980年代	超LSI	GUI・ オブジェクト指向 言語	計算機ネットワーク オンライン処理 データベース パーソナルコンピュータ ワークステーション	一般人

カーネル(kernel)

- OSの基本的なサービス
 - 資源の割付と保護
 - プログラムの実行
 - 入出力操作
 - ファイル操作

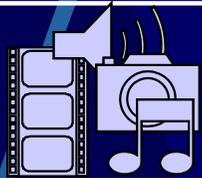
カーネル(kernel)

基本的サービスを行うOSの根幹

制御プログラムとしての役割

カーネル

ユーザ



アプリケーションプログラム



システムプログラム

エディタ, コンパイラ
リンカ, ロード 等

カーネル
(kernel)

プロセス管理, 同期と通信制御
ファイルシステム, メモリ管理
スケジューラ, 割り込み制御, 入出力制御
タイマ管理, デバイスドライバ 等



ハードウェア

機械語, 物理デバイス
マイクロプログラム 等

↑ 狭義のOS ↓

↑ 広義のOS ↓

カーネルの特徴

- カーネルの特徴
 - 割込みにより起動
 - カーネルモード(特権モード)で動作
 - アプリケーションプログラムから記憶保護
 - プログラム実行の管理
 - 計算機資源の管理
 - 特別な特権命令でアプリケーションプログラムに戻る

カーネル

- カーネルが行うサービス

- プロセス管理
- 同期と通信制御
- ファイルシステム
- メモリ管理
- スケジューラ
- 割り込み制御
- 入出力制御
- タイマ制御
- デバイスドライバ

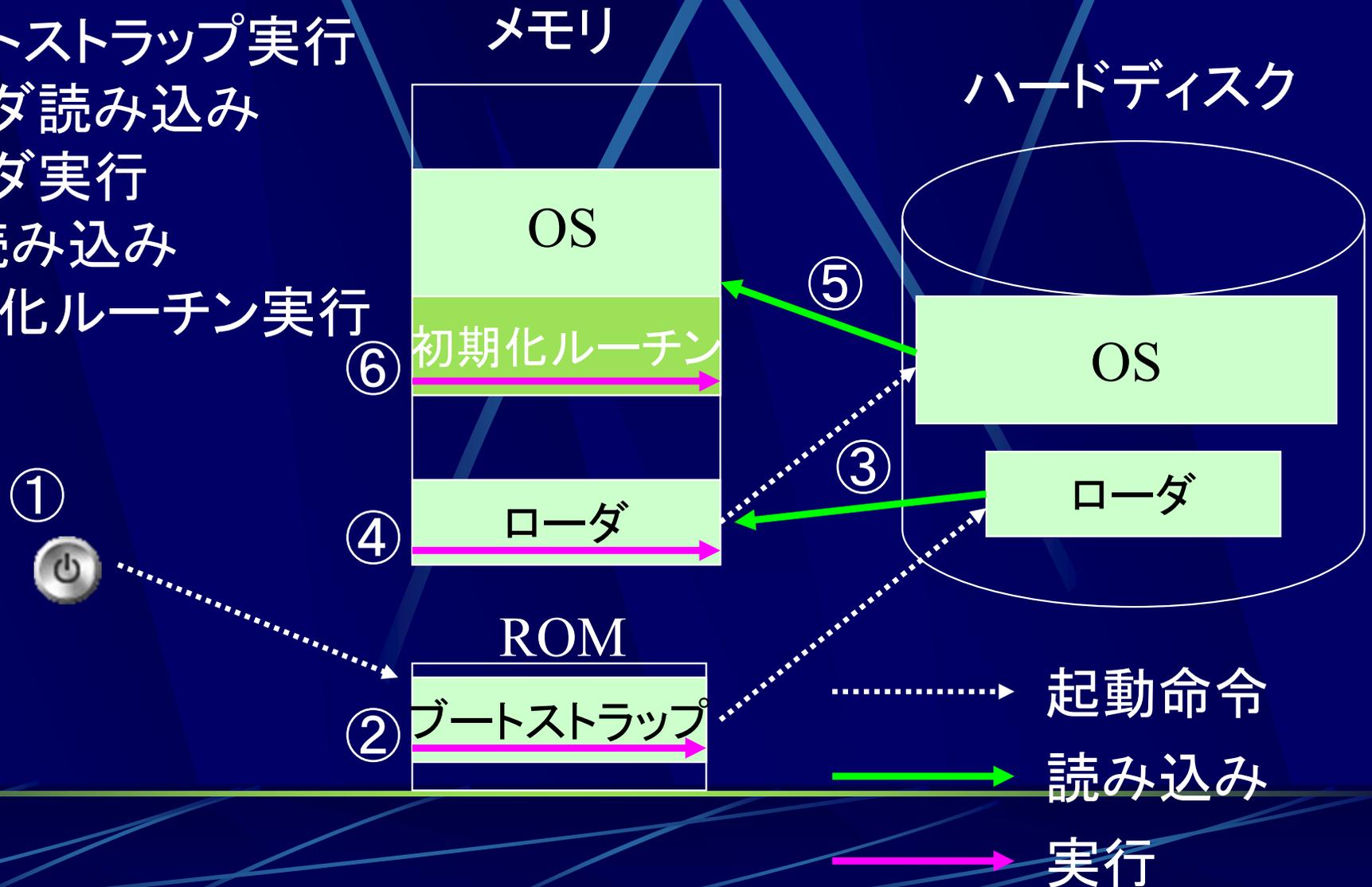
次週以降
これらについて
順次講義します

OSの起動と終了

- OSはブートストラップとローダにより起動
 - ブートストラップ(bootstrap)
 - ブートROM上に書き込まれている
 - ハードディスクからローダを読み出し, 起動する
 - ローダ(loader)
 - ハードディスク上に書き込まれている
 - OSの初期化ルーチンを実行する

OSの起動と終了

1. 電源ON
2. ブートストラップ実行
3. ローダ読み込み
4. ローダ実行
5. OS読み込み
6. 初期化ルーチン実行



OSの起動と終了

- OSの終了はシャットダウン(shutdown)プログラムで実行
 - シャットダウンプログラム
 1. 新規プロセスの生成,ログインを停止
 2. 実行中のプロセスが終了したことを確認
 3. システムプログラムのファイルを閉じる
 4. 停止

いきなり電源を切ってはいけない

課題テスト

- 毎週 GoogleClassroom上で課題テストを行う
 - 授業後～翌週の授業開始まで
- GoogleClassroomで
オペレーティングシステム
 - ⇒授業
 - ⇒その回の課題と辿る

2022-オペレーティングシステム
理工学部 情報学科 情報システムコース 2年

OS

仮想計算機

資源管理者

カスタマイズ

Meet

リンクを生成

クラスコード

bgbto2b

期限間近

提出期限の近い課題はありません

すべて表示

保存済みのお知らせ (5件)

クラスへの連絡事項を入力

石水隆
8月27日

出席カードを提出してください (9/12) <https://forms.gle/rjRvuSzQhy6qKnGs6>

クラスのコメントを追加...

石水隆 さんが新しい資料を投稿しました: 出席カード (9/12)

第3回 課題 各週課題 投稿予定: 9月27日 8:00

出席カード (10/3) 投稿予定: 10月3日 8:00

第2回 割り込みとOSの構成

第2回 講義資料 投稿日: 8月27日

第2回 課題 各週課題 投稿予定: 9月13日 8:00

出席カード (9/26) 投稿予定: 9月26日 8:00

第1回 OSの概要

第1回 講義資料 投稿日: 8月27日

第1回 課題 各週課題 期限: 9月26日

出席カード (9/12) 投稿日: 8月27日



出席カード (9/26)

投稿予定: 9月26日 8:00

第1回 OSの概要



第1回 講義資料

投稿日: 8月27日

第1回 課題

各週課題

期限: 9月26日

投稿日: 8月27日 (最終編集: 8月27日)

GoogleForm から回答してください

0
提出済み

0
割り当て済み

第1回 OSの概要
Google フォーム

課題を表示



出席カード (9/12)

投稿日: 8月27日

第1回 OSの概要

📧 .kindai.ac.jp アカウントを切り替える

このフォームを送信すると、メールアドレスが記録されます

***必須**

あなたの氏名を入力してください。 *

回答を入力

あなたの学籍番号を入力してください。(例: 2110370999) 省略形は使用しないでください。 *

回答を入力

オペレーティングシステムの役割を選べ。(3つ選択)

25 ポイント

- 資源管理者
- メール
- 制御プログラム

