

オペレーティングシステム（1 回目）

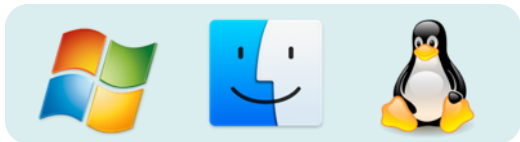
－ 概論 －

慶應義塾大学
河野健二

- Operating System (OS) の役割と仕組みを理解する
 - 汎用的な OS を対象とし, OS の基礎概念と実装技術を学ぶ
 - OS の内部解説書が読める程度までもっていきたい
- 汎用的な OS とは？
 - Windows, Mac OS, Linux など. フツウの Operating System のこと



Applications



Operating System



これがないと
アプリは動けない.
でも, OS って何?



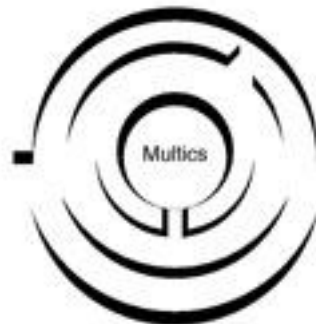
Hardware

この質問に答えるのが, この講義

■ 長い研究と開発経験の集大成

- 1960 年代に開発された Multics という OS からの蓄積

<https://www.multicians.org/>



■ 現在もどんどん進化を続けている

- Linux の例: たった 1 年で...

コード追加: 3,385,121 行, コード削除: 2,512,040 行.



As of this New Year's Eve afternoon, the Linux kernel saw 74,974 commits this year that added 3,385,121 lines of code and removed 2,512,040 lines.

For as impressive as seeing almost 75k commits in a single year to an open-source project, it's not actually a record high. Last year in fact saw 80,725 commits that added 3.9 million lines and removed 1.3 million lines... A much

なぜオペレーティングシステムを学ぶのか？

■ OS がわかっていないと・・・

- コンピュータはブラックボックスのまま



Applications

ブラックボックス

Operating System

Hardware

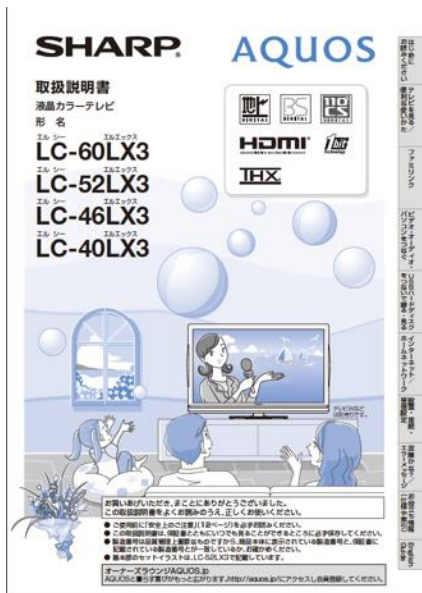
OS が見えてきて、
はじめてコンピュータの
仕組みが見える

■ 結果的に・・・

- コンピュータは使いこなせない
- 高度なプログラミングはできない
- 時代の進歩に追従できない
 - ◆ ハードウェア & ソフトウェアは
どんどん進化する
 - ◆ その進化を支えるのが OS

オペレーティングシステムとは？

- コンピュータを動かすための基盤ソフトウェア
 - OS がないとアプリケーションは動かない
 - コンピュータがあれば, 必ず OS が動いている



謝辞

本機には以下のフリーソフトウェアコンポーネントが組み込まれています。

- linux kernel
- module-init-tools
- glibc
- DirectFB
- OpenSSL
- 7lih
- coreutils
- jpeg
- libpng
- SQLite
- LVM2
- hash

オペレーティングシステムとは？

- コンピュータを動かすための基盤ソフトウェア
 - OS がないとアプリケーションは動かない
 - コンピュータがあれば, 必ず OS が動いている

 RIOT GAMES

OUR

OS SCHEDULER



AUG 31, 2020

VALORANT'S 128-TICK SERVERS

During our time monitoring the game server host, we saw an interesting pattern v
at around 90-96% usage but never reach 100% - even when the host was loaded
games it should be able to host. We suspected the OS scheduler was the cause,
profiling tool, Adaptive Optimization, along with the *perf* utility of Linux to dig into
Utilizing Linux also meant we had the chance to review the source code for the s

オペレーティングシステムとは？

- コンピュータを動かすための基盤ソフトウェア
 - OS がないとアプリケーションは動かない
 - コンピュータがあれば, 必ず OS が動いている

Nintendo Switch system software

Article [Talk](#)

From Wikipedia, the free encyclopedia



This article's [lead section](#) may be too short to adequately [summarize](#) the article. [Consider expanding the lead to provide an accessible overview of the article.](#) (August 2023)

The **Nintendo Switch system software** (also known by its codename **Horizon**)^[3] is an updatable [firmware](#) and [operating system](#) used by the [Nintendo Switch video game console](#). It is based on a proprietary [microkernel](#). The UI includes a HOME screen,



じゃあ、オペレーティングシステムって何をやってるの？

- 「オペレーティングシステム」の定義

- “合意のとれた明確な定義はない” という合意が取れている



でも、「何をするものなのか」という合意は取れている

- 裸のコンピュータを抽象化(あるいは仮想化)し,

- 使いやすいコンピュータとして見せること

- “使いやすい” == “プログラミングがしやすい”

■ マニュアルの車：

- クラッチの役割, ギアの役割を知らないといけない
- 運転状況に応じてクラッチとギアを操作しないといけない



クラッチとギアを「抽象化」



■ オートマの車：

- 運転が楽
 - ◆ アクセルとブレーキの操作だけでよい
- 運転手からはクラッチとギアの存在が隠蔽されている



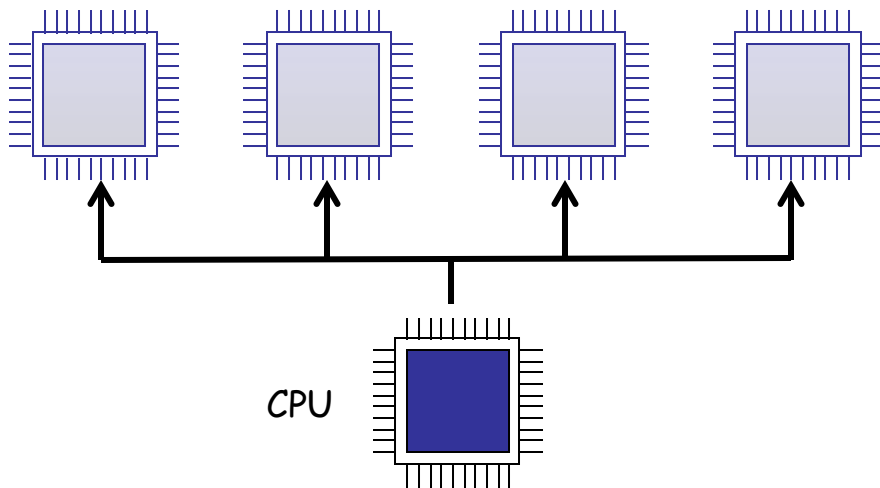
- ハードウェアのことを知らなくても、プログラミングできるようにすること



具体的には・・・

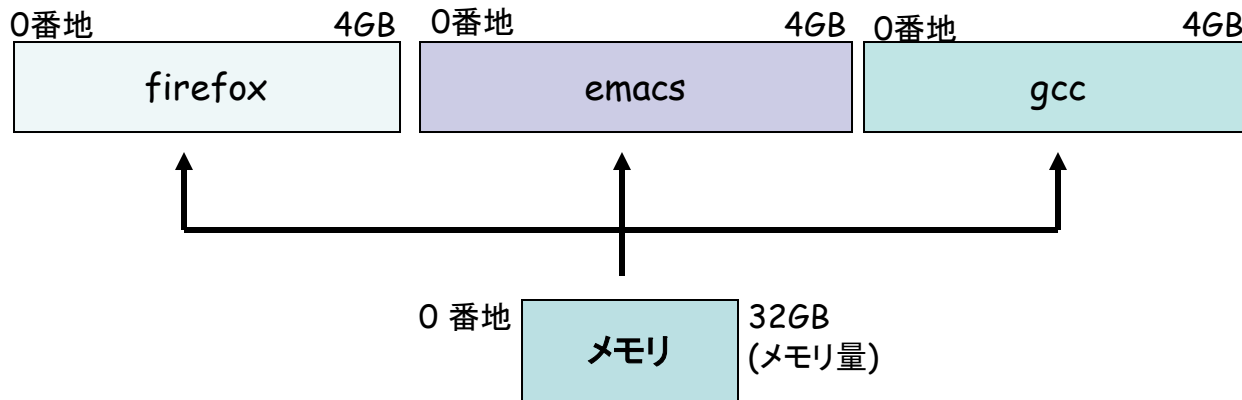
- ハードウェアの詳細を隠蔽する
 - ハードウェアの細かい仕様を知らなくてもよい
- ハードウェアによる制約を緩める
 - ハードウェアによる物理的な限界からソフトウェアを解放

- 物理的なイメージ
 - ひとつの CPU はひとつのプログラムしか実行できない
- OS が作り出すイメージ
 - ひとつの CPU で複数のプログラムが同時に実行できる
 - それぞれのプログラムが専用の CPU を持つようにみえる



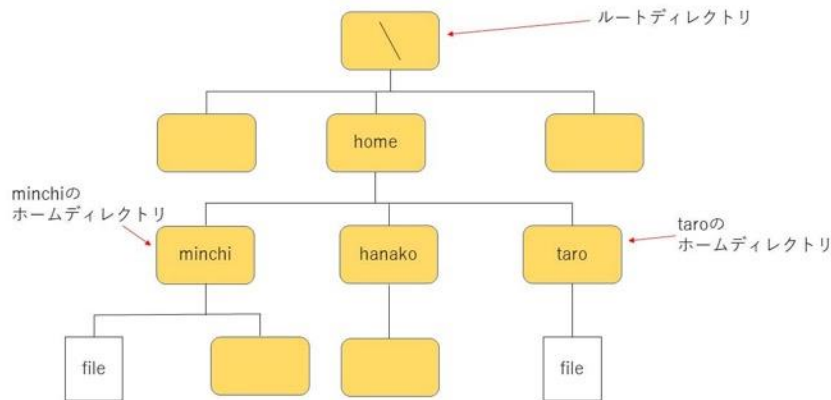
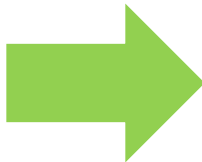
- 物理的なイメージ
 - アドレスのついた記憶場所
 - コンピュータに搭載している分量しか使えない
- OS が作り出すイメージ
 - 実行中のプログラムそれぞれに専用の記憶場所
 - コンピュータに搭載している分量を超えてメモリが使える

32 bit CPU では, 4GB
64 bit CPU では, 256TB



ハードディスクの抽象化

- 物理的なイメージ
 - 固定長の記憶領域が並んだもの
- OS が作り出すイメージ
 - ファイル, ディレクトリ (フォルダ) など



- 悪いことができないようにすること
 - OS の提供する重要な機能のひとつ
- 例：
 - あるプログラムがCPUを独占できない
 - 他のプログラムのメモリを破壊できない
 - 通信内容を横取りできない
 - 権利のない人がコンピュータを利用できない
 - 他人のファイルを勝手に読み書きできない
- 操作ミスだろうが悪意があろうが、できないようにする

ところで“汎用的”ではない OS って？

■ 汎用的な OS:

- 頼まれた仕事は，“いつか”終わらせることしか保証しない
- 時間保証はしない

そのうち
やります



いつか必ず
やります

■ 組み込みリアルタイム OS:

- エンジン制御などの極小コンピュータ向け
- 頼まれた仕事は，“締切”までに終わらせる
 - ◆ 時間保証する

- 一見、こっちの方が凄そうだが...

- ◆ 時間保証のため、標準 OS のもつ面白い機能はない
- ◆ (汎用 OS から見れば) おもちゃみたいな感じ

機能不足のため、
最近は Linux で
代替すること多い



えっへん

講義予定 (進捗に応じて変更あり)

10/3	OS 概論	11/28	相互排除と同期 (2)
10/10	I/O デバイスと割込み	12/5	仮想記憶 (1)
10/17	プロセスとスレッド	12/12	仮想記憶 (2)
10/24	スケジューリング	12/19	プログラミング課題 (2)
10/31	カーネルの保護	12/26	ファイルシステム (オンデマンド)
11/7	相互排除と同期 (1) (遅れるかも)	1/9	質問受付
11/14	プログラミング課題 (1)	1/16	繰り上げ試験

- 資料の配布・課題の提出など
 - すべて K-LMS を利用
 - 大事なアナウンスなども K-LMS を利用
 - (講義時間外の) 質問なども「受信箱」を利用

- 繰り上げ試験を行う. 比重は 80%
 - 持ち込み不可
 - 冬休み前に過去問を K-LMS にアップ予定

- プログラミング課題もある. 比重は 20%
 - 講義で習ったことを試してもらう
 - ◆ プログラムを書くというより, 習ったことを実体験してもらうのが目的

 - 標準的な課題はやりやすい形で出題する
 - ◆ 誘導にしたがって作業すれば, 講義の理解が深まるようにする

 - ボーナス課題はやりがいのあるものを出題する
 - ◆ 普通の人はやらなくてよい

- 拙著
「オペレーティングシステムの仕組み」
 - 朝倉書店,
2007年10月出版
 - amazon.co.jp で 5 つ星がついています
- ちょっと古くなっちゃったので,
買う必要はありません



- ***Operating Systems: Three Easy Pieces***

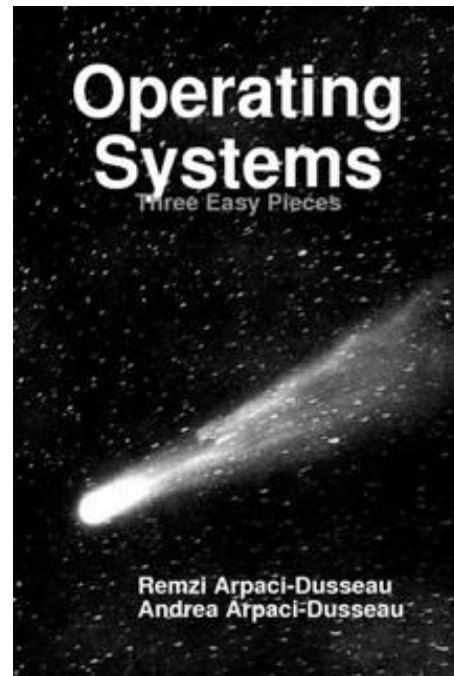
- Remzi & Andrea Arpaci-Dusseau（著）

- 無料で読める

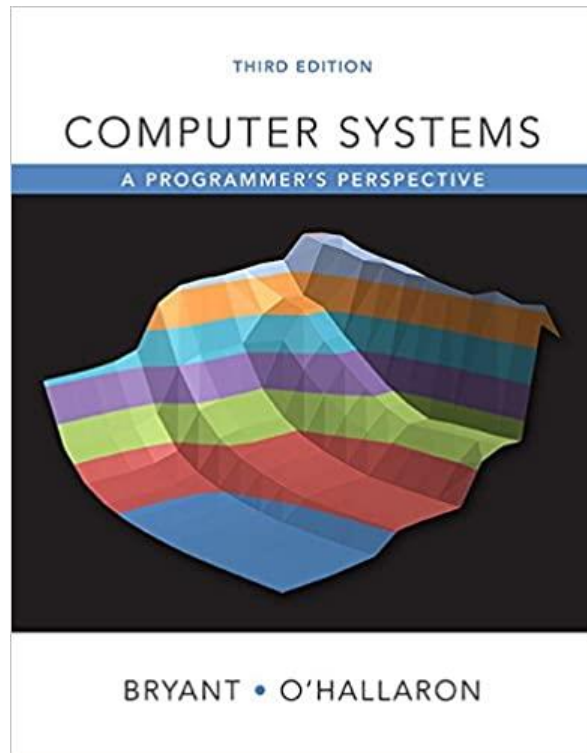
- <http://pages.cs.wisc.edu/~remzi/OSTEP/>

- OS 研究の第一人者の教科書

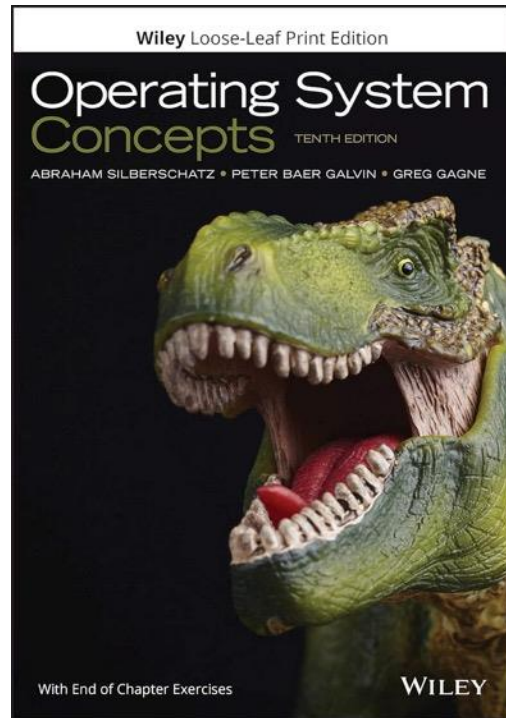
- 基本から最近の話題まで
 - とてもわかりやすく親切
 - 英語そのものは簡単



- ***Computer Systems:
A Programmer's Perspective***
 - Randal Bryant, David O'Hallaron
 - ISBN: 978-0134092669
- プログラマにとって必要な
OS/アーキテクチャについての
知識を網羅的に解説
- 和訳もあり
(訳者がいまいちかも)



- ***Operating System Concepts*** 10th edition
 - A. Silberschatz, P. Galvin, Greg Gagne (著)
 - ISBN: 978-1119800361
- 国際的に評価の高い標準的な教科書
 - 2nd Edition が評判が高かった
 - 一時期, 内容が多くなりすぎた
 - 10th edition では改善された模様
- 古い版の和訳もあるらしい



- xv6 がおすすめ. MIT で開発された教育用 OS
 - シンプルな構造で読みやすい
 - C 言語 + x86 のアセンブラで 10,000 行程度
 - <https://pdos.csail.mit.edu/6.828/2019/xv6.html>

6.828: Operating System Engineering

ScheduleClass ~ Labs ~ xv6 ~ ReferencesPlazza

2018

Xv6, a simple Unix-like teaching operating system

Introduction

Xv6 is a teaching operating system developed in the summer of 2006 for MIT's operating systems course, [6.828: Operating System Engineering](#). We hope that xv6 will be useful in other courses too. This page collects resources to aid the use of xv6 in other courses, including a commentary on the source code itself.

History and Background

For many years, MIT had no operating systems course. In the fall of 2002, one was created to teach operating systems engineering. In the course lectures, the class worked through [Sixth Edition Unix \(aka V6\)](#) using John Lions's famous commentary. In the lab assignments, students wrote most of an exokernel operating system, eventually named Jos, for the Intel x86. Exposing students to multiple systems—V6 and Jos—helped develop a sense of the spectrum of operating system designs.

V6 presented pedagogic challenges from the start. Students doubted the relevance of an obsolete 30-year-old operating system written in an obsolete programming language (pre-K&R C) running on obsolete hardware (the PDP-11). Students also struggled to learn the low-level details of two different architectures (the PDP-11 and the Intel x86) at the same time. By the summer of 2006, we had decided to replace V6 with a new operating system, xv6, modeled on V6 but written in ANSI C and running on multiprocessor Intel x86 machines. Xv6's use of the x86 makes it more relevant to students' experience than V6 was and unifies the course around a single architecture. Adding multiprocessor support requires handling concurrency head on with locks and threads (instead of using special-case solutions for uniprocessors such as enabling/disabling interrupts) and helps relevance. Finally, writing a new system allowed us to write cleaner versions of the rougher parts of V6, like the scheduler and file system. 6.828 substituted xv6 for V6 in the fall of 2006.

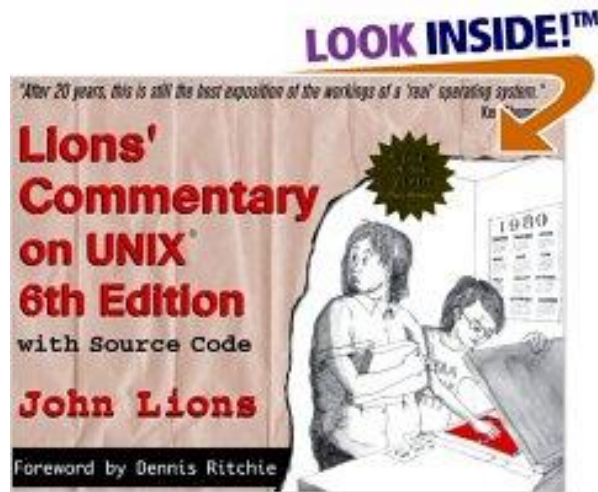
Xv6 sources and text

■ *Lions Commentary on UNIX 6th Edition*

- John Lions（著）

■ 押しも押されぬ古典的名著

- UNIX のソースに
1 行 1 行丁寧に解説
- Ritchie と Thompson の
芸術的な C コードが魅力



■ この本の内容を現代的に焼き直したのが xv6

OS の実装について（その他）

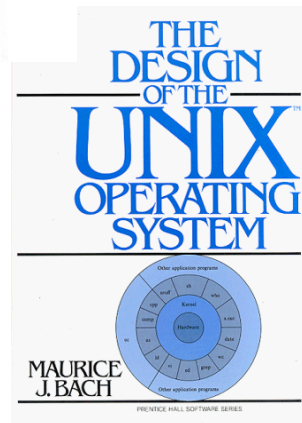
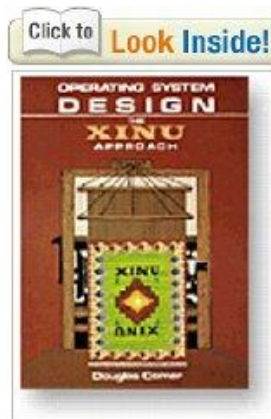
- Douglas E. Comer
Operating System Design: The XINU Approach

XINU という教育用の簡単な OS を用い, その内部構造を解説. 簡潔でわかりやすい

- Maurich J. Bach
The Design of the UNIX Operating System

UNIX の内部構造についてソースコードを交えて解説.

UNIX の基本構造の理解に最適. 著者の名をとって俗に Bach 本という



- OS の役割
 - 裸のコンピュータを抽象化し, 使いやすいコンピュータとして見せること
- OS を学ぶ意義
 - コンピュータシステムの動作原理を理解する
- OS の機能と抽象化
 - CPU: プロセス・スレッドの提供
 - メモリ: 仮想アドレス空間, 仮想記憶
 - 二次記憶装置: ファイルシステムの提供
 - 保護機能