

# デバイスの仮想化



## 入出力デバイス管理

- 共有のための資源管理の機能
  - 排他管理
    - 入出力機器を占有して使う必要がある
  - 効率向上
    - ディスクスケジューリング  
バッファリング・ブロッキング・プリフェッチ
- 仮想化の機能



前の節



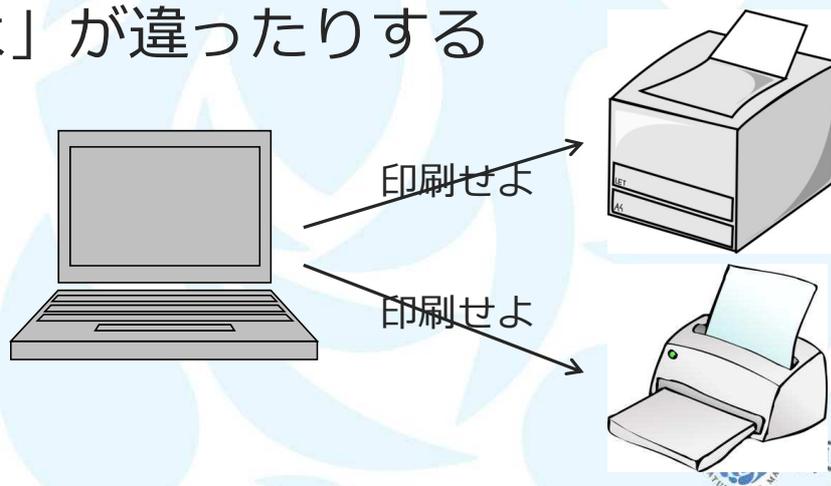
この節



# 仮想化とはどういう考え方が

## 仮想化の考え方

- (前提) 入出力機器は機種によって操作方法やデータ形式が違う
- 例：A社のプリンタ vs B社のプリンタで「印刷せよ」が違ったりする



## 仮想化の考え方

- (前提) 入出力機器は機種によって操作方法やデータ形式が違う
  - 例：A社のプリンタ vs B社のプリンタで「印刷せよ」が違ったりする

↓

- 機種ごとに違いを理解して違うプログラムを書かなければならない

## 仮想化の考え方

- (前提) 入出力機器は機種によって操作方法やデータ形式が違う
  - 例：A社のプリンタ vs B社のプリンタで「印刷せよ」が違ったりする

↓

- 機種ごとに違いを理解して違うプログラムを書かなければならない
  - ⇒ 同じprintコマンドで済ませたい

# 仮想化の考え方

- 入出力デバイスの仮想化は  
いろいろな違いを共通化して  
(プログラムからの) 同じ操作を提供

6

# 仮想化の考え方

- 入出力デバイスの仮想化は  
いろいろな違いを共通化して  
(プログラムからの) 同じ操作を提供
  - 機器の種類ごとの操作方法 (命令) の違い
    - A社のプリンタXと、B社のプリンターY
    - プリンター出力と、画面出力と (多少ウソだが)
    - ハードディスクと、USBメモリと (これもウソだが)

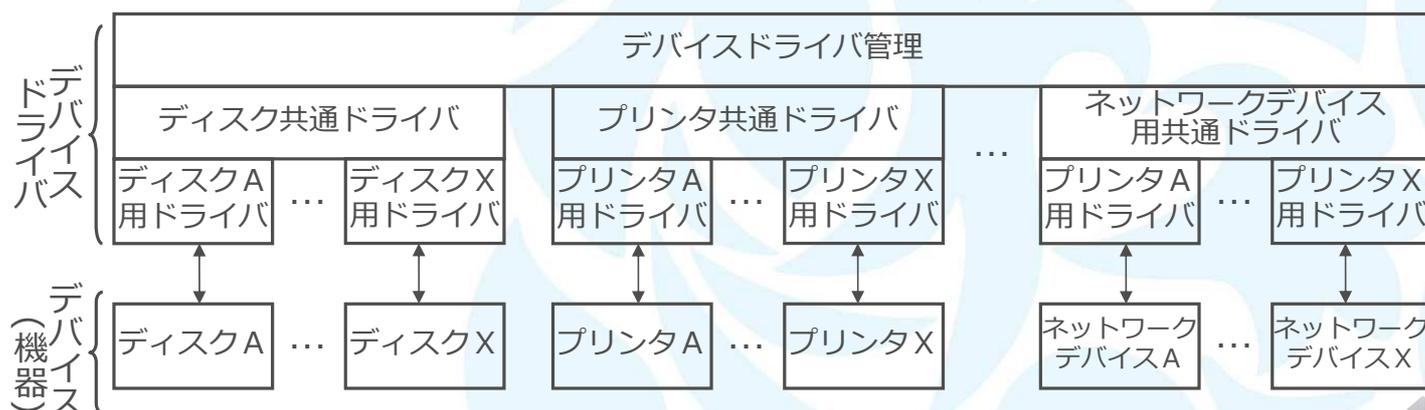
7

# 仮想化の考え方

- 入出力デバイスの仮想化は  
いろいろな違いを共通化して  
(プログラムからの) 同じ操作を提供
  - 機器の種類ごとの操作方法 (命令) の違い
  - 接続方法の違い (ハードウェア授業、教p56~)
    - 細かい話はハードウェアの授業を思い出して下さい

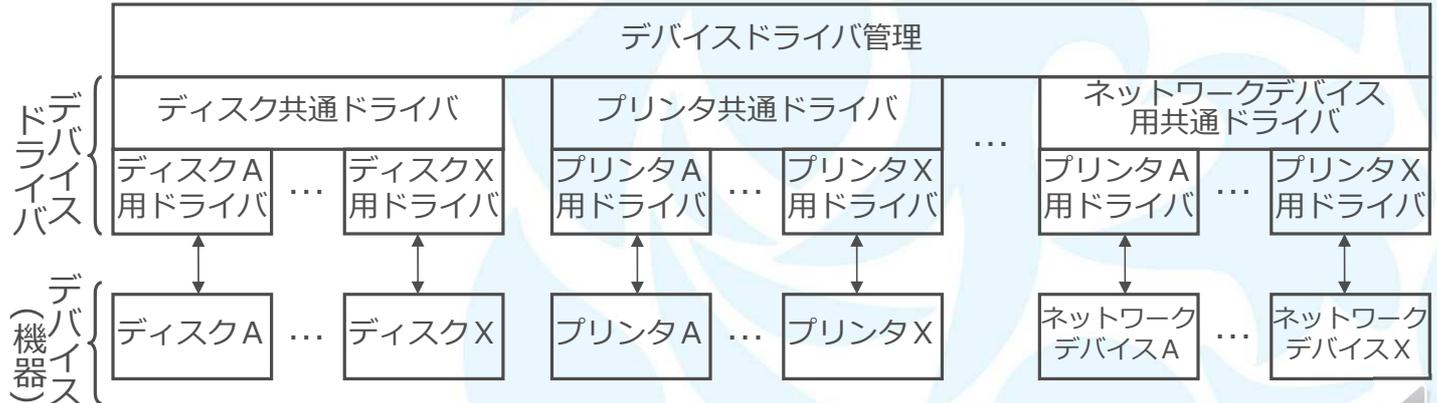
## 仮想化はいくつかレベルがある

- 教科書p61の図を見ると



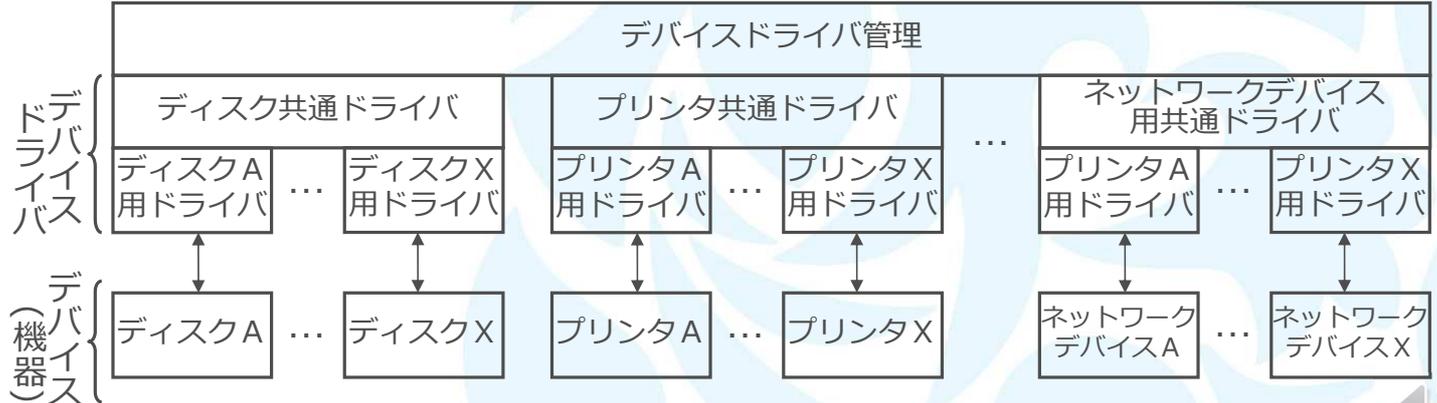
# 仮想化はいくつかレベルがある

- 教科書p61の図を見ると  
2段構えになっている  
共通ドライバと、個々の機器のドライバ  
OSによって階層数・分類の仕方は違う



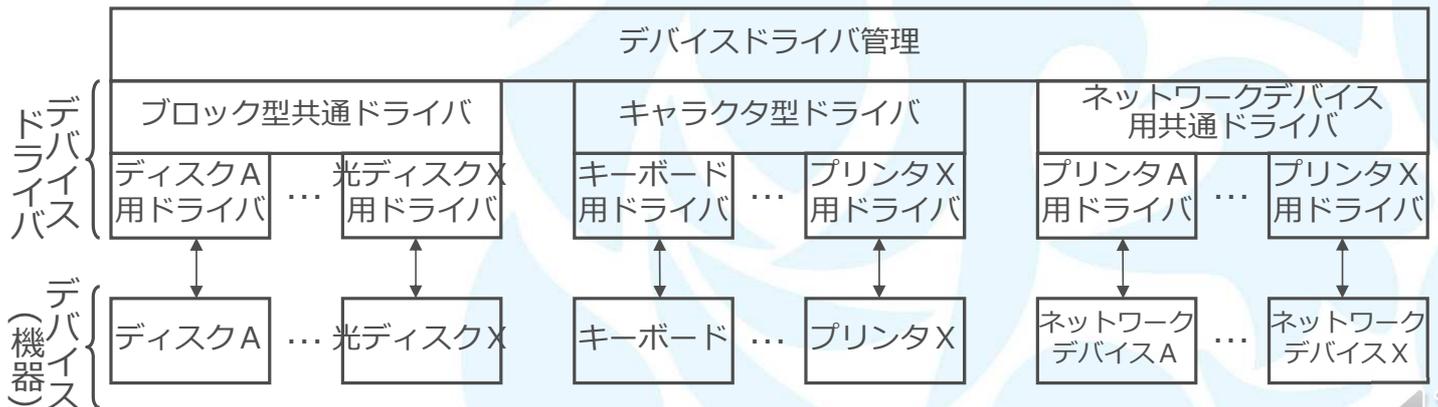
# 仮想化はいくつかレベルがある

- 個々のドライバは、機器を付けた時インストール  
<X社製のYモデルのプリンタのドライバ>
- 共通ドライバはOSの一部として供給



# 仮想化はいくつかレベルがある

- Linuxでは共通ドライバの分け方が  
 文字型 ~ バイト単位の入出力  
 ブロック型 ~ 多数バイトブロックで入出力  
 ネットワーク型 ~ パケットで入出力



12



東邦大学

## デバイスドライバは

- (全デバイスに共通した) 抽象的な操作をそれぞれの機器 (固有) の操作に変換する
- イメージ: Writeという抽象的な操作を、プリンタの「印刷せよ」に変換する

13



東邦大学

# デバイスドライバは

- (全デバイスに共通した) 抽象的な操作をそれぞれの機器 (固有) の操作に変換する
  - イメージ: Writeという抽象的な操作を、プリンタの「印刷せよ」に変換する
- 抽象的な操作は (基本的に) Open, Read, Write, Close の4種類

14

## 基本操作

- Open: デバイスを動作させる準備をする
  - デバイスを占有し、必要な初期化をする  
動作に必要な作業領域を確保する、 など

15

# 基本操作

- Open: デバイスを動作させる準備をする
  - デバイスを占有し、必要な初期化をする  
動作に必要な作業領域を確保する、 など
- Read・Write: デバイスを読む・書く
  - デバイスにデータを送る・デバイスから受取る

16



# 基本操作

- Open: デバイスを動作させる準備をする
  - デバイスを占有し、必要な初期化をする  
動作に必要な作業領域を確保する、 など
- Read・Write: デバイスを読む・書く
  - デバイスにデータを送る・デバイスから受取る
- Close: 終了処理
  - デバイスを解放し、領域を解放、終了処理する

17



# 基本操作

- Open: デバイスを動作させる準備をする
- Read・Write: デバイスを読む・書く
- Close: 終了処理
  
- 基本的に、この4操作だけ  
(実はこれでは困ることもあるが)

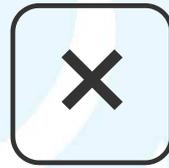
18

## ここまでのまとめ

- 入出力デバイスの管理の中で、  
仮想化の機能を概観した
- 仮想化は  
個々の機器・機種の違いを意識しないで  
利用できるためのサービスである
  - 基本的な動作としてOpen/Read/Write/Close
  - デバイスドライバによって実現

19

デバイス仮想化の考え方が  
掴めましたか？



↓  
次へ