

オペレーティングシステム

第2回

割り込みとOSの構成

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

E館3階E-331 内線5459

takasi-i@info.kindai.ac.jp

1



2

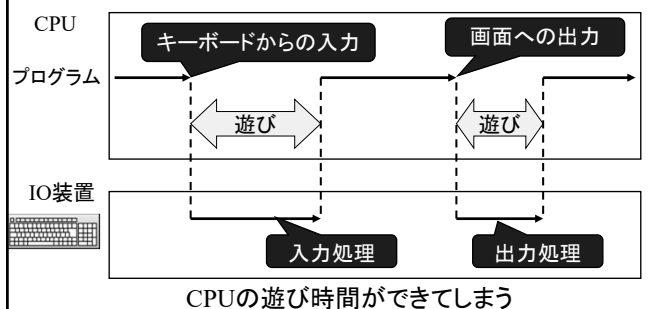


3



4

プログラムの実行中の動作



5

単一プログラムの問題点

単一プログラムだとCPUの遊び時間が大きくなる

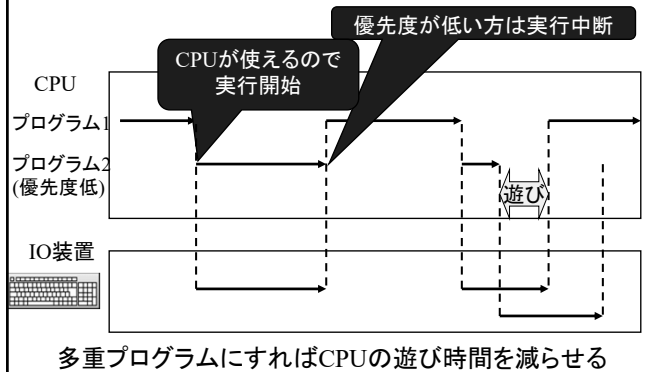
多重プログラムにする

多重プログラム

- 複数のプログラムを並行して実行させる

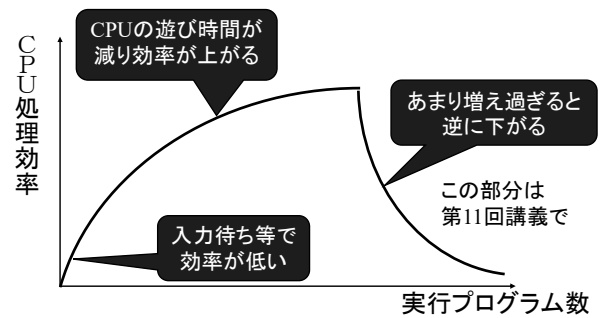
6

多重プログラムの実行中の動作



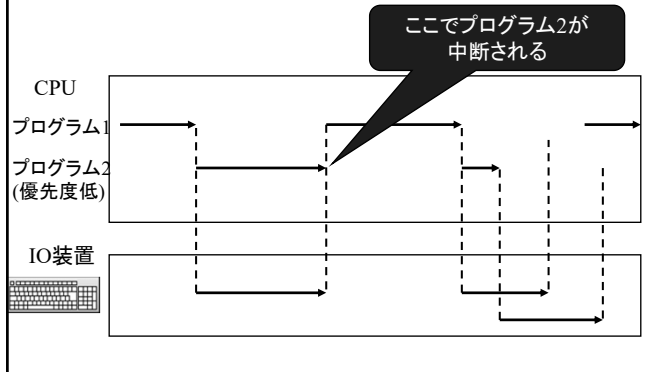
7

実行プログラム数と処理効率



8

多重プログラムの実行中の動作

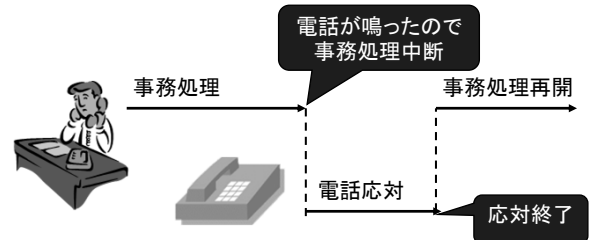


9

割り込み(interrupt)

● 割り込み(interrupt)

- 実行中の処理を中断して特別な処理をする

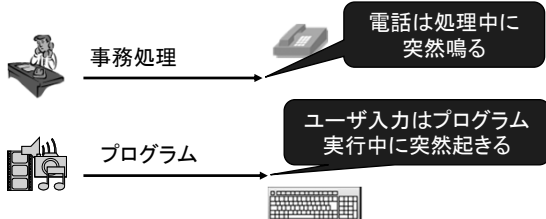


10

割り込みの発生

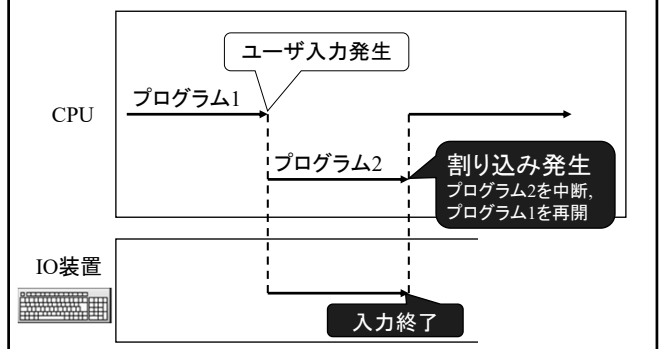
● 割り込みは非同期に発生する

- 電話はいつ鳴るかは分からない
- ユーザ入力はいつ完了するかは分からない



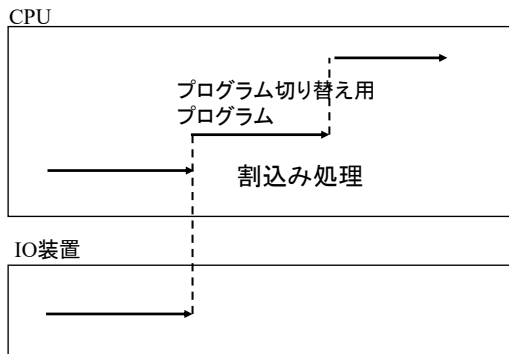
11

割り込み



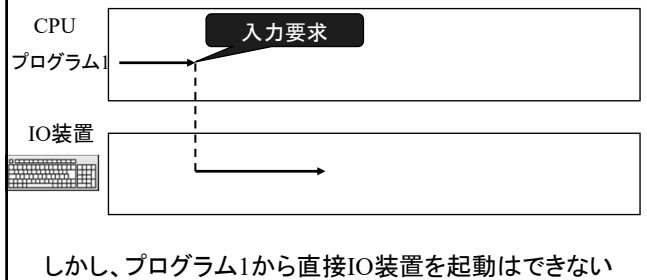
12

割り込み



13

プログラム実行中の入力要求



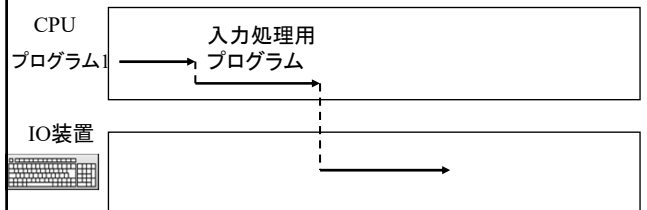
14

プログラム実行中の入力要求

- アプリケーションプログラムが直接IO装置を起動することはできない
 - どのプログラムがIO装置を使うかの管理
 - 実行プログラムの切り替え
- ↓
- これら処理するプログラムが必要

15

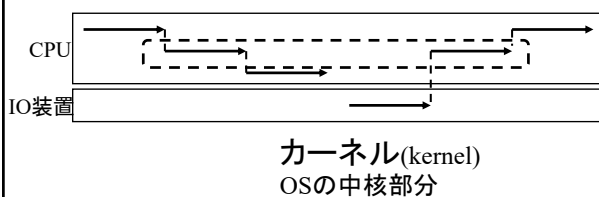
プログラム実行中の入力要求



16

特別な処理を行うプログラム

- 特別な処理を行うプログラム
 - プログラム切り替え用プログラム
 - 入出力処理用プログラム



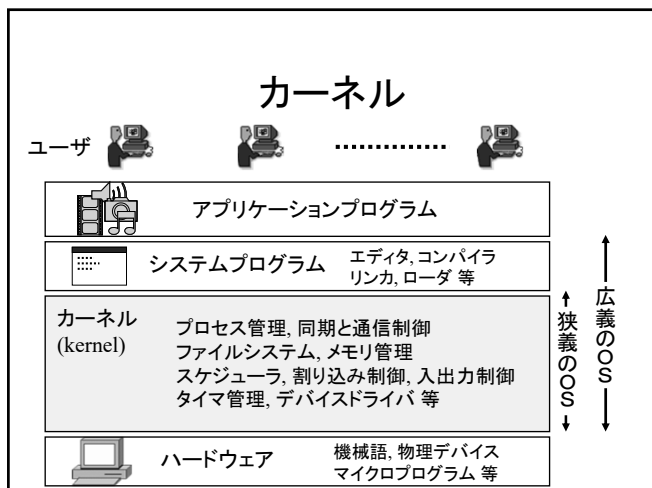
17

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

18



19

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

20

カーネルモード, 特権モード (kernel mode, privileged mode)

- 計算機全体に影響を与える命令を実行できるモード
 - 資源の管理, 制御

例: メモリ管理

アプリケーションプログラム

→

メモリ

アプリケーションプログラムが直接メモリアクセスするのは禁止

他のプログラムの領域を書き換えられる危険

アプリケーションプログラムにメモリ管理という余計な負荷

メモリ管理はOSが行う

21

資源の管理, 制御

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

OS

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

↓

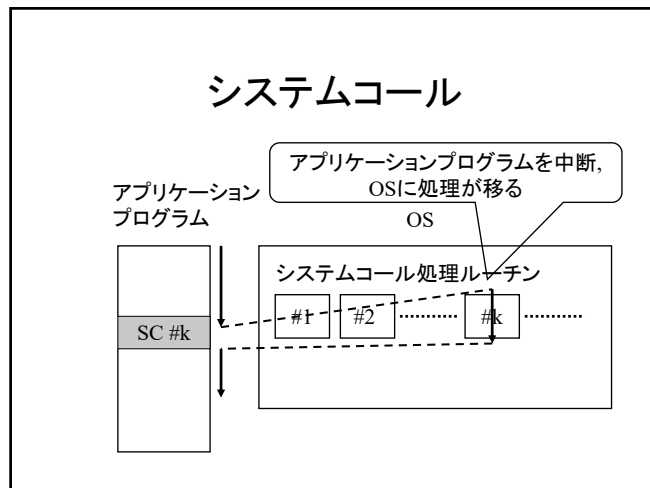
アプリケーションを停止する

22

システムコール(system call)

- アプリケーションプログラムからOSへのサービス要求
 - 入出力をしたい
 - 他のプログラムを起動・停止したい
 - ファイルに読み・書きしたい
- 手続き呼び出しの形式を取る

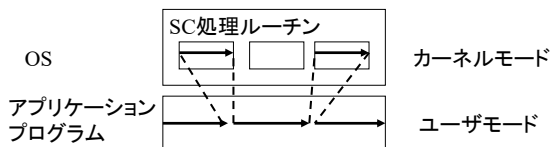
23



24

カーネルモード, ユーザモード (kernel mode, user mode)

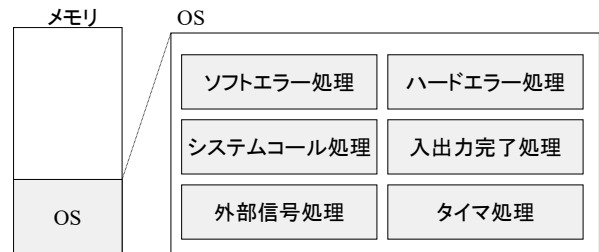
- カーネルモード
 - システムコール等の割り込みにより発生, 各種割り込み処理ルーチン実行中
- ユーザモード
 - アプリケーションプログラムを実行中



25

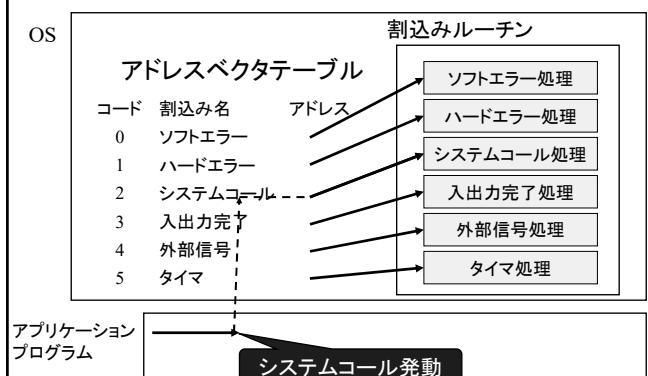
割り込み

- メモリのOS領域内に割り込みの処理ルーチン



26

割り込み



27

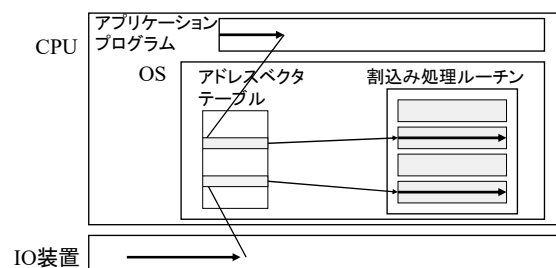
カーネルの構成要素

1. 割り込み制御
2. 入出力制御
3. タイマ管理
4. 記憶管理
5. CPUスケジューラ
6. プロセス管理
7. 同期と通信制御
8. ファイルシステム

28

カーネルの構成要素 割り込み制御

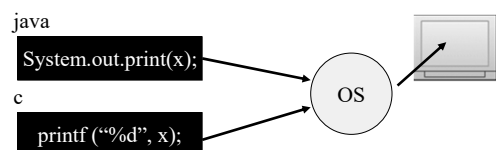
1. 割り込み制御
 - 割り込み要因の解析と処理ルーチンへの分岐



29

カーネルの構成要素 入出力制御

2. 入出力制御
 - 入力動作のスケジューリング
 - 入出力装置の仮想化

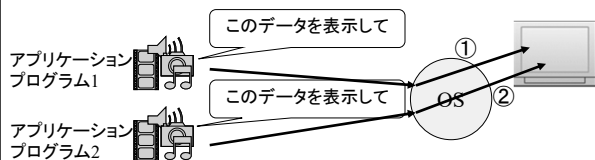


30

カーネルの構成要素 入出力制御

2. 入出力制御

- 入力動作のスケジューリング
- 入出力装置の仮想化

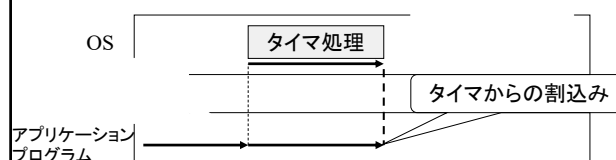


31

カーネルの構成要素 タイマ制御

3. タイマ制御

- 時刻の管理
- 時間経過の監視, 通知など

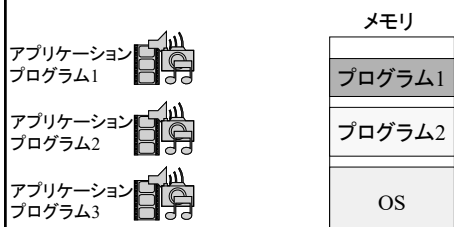


32

カーネルの構成要素 記憶管理

4. 記憶管理

- 主記憶管理: メモリ割り付け
- 仮想記憶管理: ページング, 動的置き換え

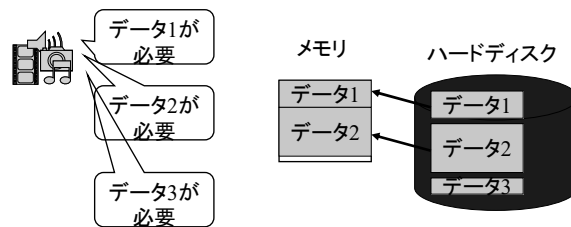


33

カーネルの構成要素 記憶管理

4. 記憶管理

- 主記憶管理: メモリ割り付け
- 仮想記憶管理: ページング, 動的置き換え

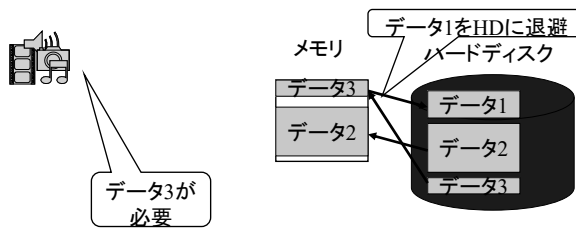


34

カーネルの構成要素 記憶管理

4. 記憶管理

- 主記憶管理: メモリ割り付け
- 仮想記憶管理: ページング, 動的置き換え

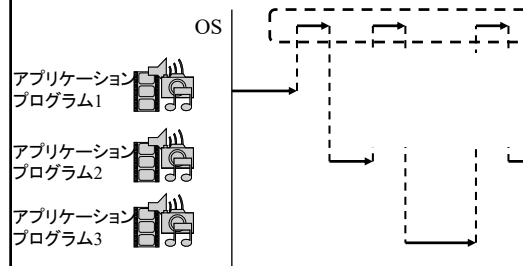


35

カーネルの構成要素 CPUスケジューラ

5. CPUスケジューラ

- CPUのプロセスへの割り付けを管理

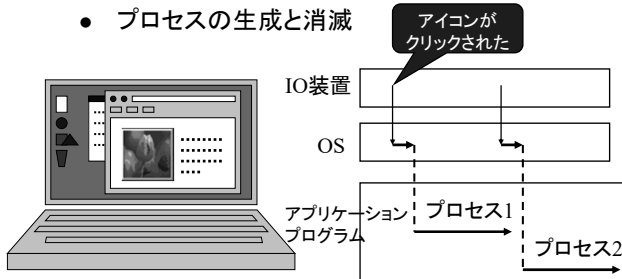


36

カーネルの構成要素 プロセス管理

6. プロセス管理

- プロセスの生成と消滅

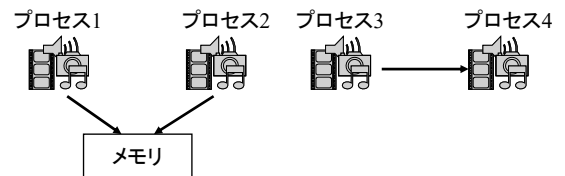


37

カーネルの構成要素 同期と通信制御

7. 同期と通信制御

- 並行プロセス間の協調処理
 - プロセス間同期: 排他制御, read-writeの同期
 - プロセス間通信: メッセージ転送



38

カーネルの構成要素 ファイルシステム

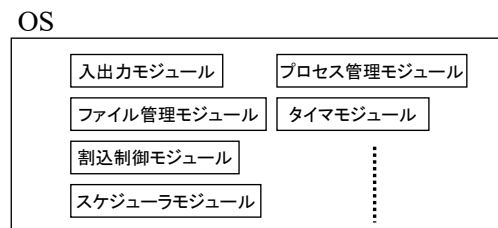
8. ファイルシステム

- ファイル制御の規定と提供
 - ファイル構造: バイト列/レコード列, キー
 - アクセス法: 順アクセス, ランダムアクセス
- ディレクトリ管理
- ファイル保護
 - 領域割り付け
 - 一貫性制御

39

OSの実現法

- OSは多数のモジュールから構成される



40

OSの実現法 モジュール化

- モジュール化の基準

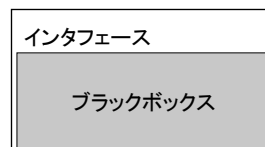
- 情報隠蔽 (information hiding)
- 方針と機構の分離
- 階層化 (Layering)

41

OSの実現法 情報隠蔽

- モジュール内部の情報を隠蔽
- インタフェースのみ公開

モジュール



- モジュール内で
- どう処理をしているか
 - ローカル変数
 - ローカルメソッド
- を隠蔽する

- モジュール内の変更が他のモジュールに影響しない
- システム全体の見通しが立ちやすい

42

OSの実現法 抽象データ型(abstract data type)

- 抽象データ型(abstract data type)
 - 複数の手続きによりデータ型を定義

例：時計モジュール

時計モジュール

- 現時刻を教えてくれる
- 時間を計れる
- 決まった時刻にアラーム

43

OSの実現法 方針と機構の分離

- 方針を決定する部分と実際の処理を行う部分を別のモジュールにする

例：CPUスケジューラとディスパッチャ

CPUスケジューラ

- スケジューリングアルゴリズムの実現
- プロセッサを割り当てるプロセスを選択

ディスパッチャ

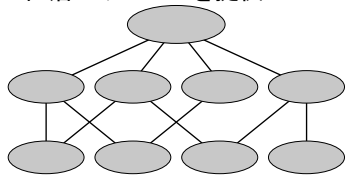
- プロセッサ割り付けの際のレジスタの退避、回復

おのおのの変更、拡張を独立に行える

44

OSの実現法 階層化(layering)

下位層のサービスを利用
上位層にサービスを提供



メインの手続き

サービス手続き
(システムコールを実行)

ユーティリティ手続き
(サービス手続きの支援)

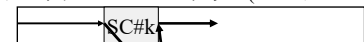
他の層の実装を知る必要は無い
(インタフェースのみ知っていればよい)
実装を考慮した階層化が必要

45

OSの実現法 モノリシックカーネル(monolithic kernel)

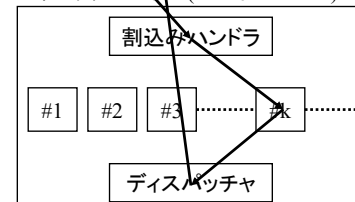
- OSの機能を全て取り込んだカーネル

アプリケーションプログラム(ユーザモード) monolithic :



一枚岩の、単層の

モノリシックカーネル(カーネルモード)



46

OSの実現法 モノリシックカーネル

OSの機能を全て取り込む = 巨大なカーネル

例：UNIX

- 機能強化に伴って巨大化

モノリシックカーネルの問題点

- カーネルはメモリに常駐
⇒ 不必要にメモリを占有
- 1つの巨大なプログラム
⇒ 機能変更、拡張に対する柔軟性に欠ける

47

OSの実現法 マイクロカーネル(micro kernel)

カーネル+システムサーバで構成

- カーネル

- 割り込み処理、スケジューリング、プロセス間通信、メモリ管理等

- システムサーバ (system server)

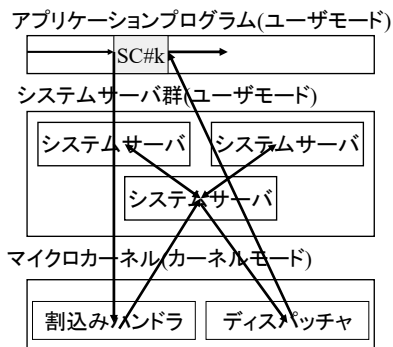
- メモリマネージャ、ファイルサーバ、ネームサーバ、デバイスドライバ等

カーネルは必要最低限の機能のみを持つ

クライアント・サーバ型

48

OSの実現法 マイクロカーネル



49

OSの実現法 マイクロカーネル

- 長所
 - 見通しの良いシステム設計
 - 変更, 拡張の容易性
 - 分散システムに適合
- 短所
 - プロセス間通信の多用によるオーバーヘッド

50

まとめ:OSの概要

- 多重(マルチ)プログラム
 - 複数のプログラムを見かけ上同時に実行
 - CPUの遊び時間を減らせる
- 割り込み
 - 実行中のプログラムを中断して特別な処理をする
 - カーネル(特権)モードで割り込み処理ルーチン
- システムコール
 - アプリケーションプログラムからOSへのサービス要求

51

まとめ:カーネルの構成要素

- | | |
|--|--|
| <ol style="list-style-type: none"> 1. 割り込み制御 2. 入出力制御 3. タイマ管理 4. 記憶管理 5. CPUスケジューラ 6. プロセス管理 7. 同期と通信制御 8. ファイルシステム | <ul style="list-style-type: none"> ● カーネルの実現 <ul style="list-style-type: none"> ● モジュール化 ● 情報隠蔽抽象データ型 ● 階層化 ● モノリシックカーネル or マイクロカーネル |
|--|--|

52

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

53