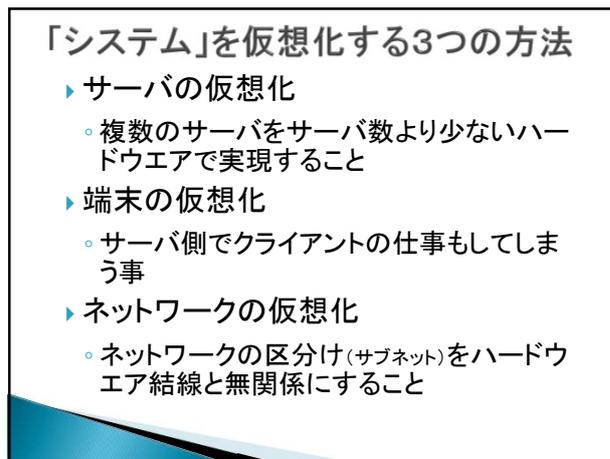
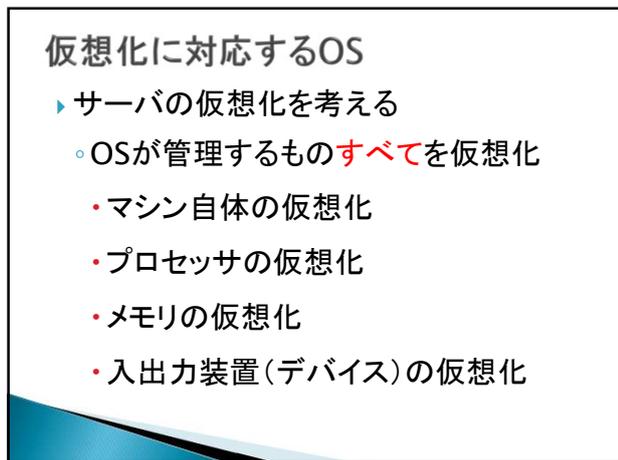




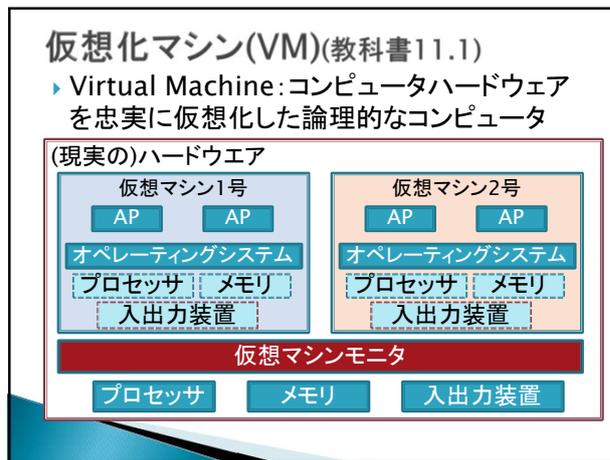
1



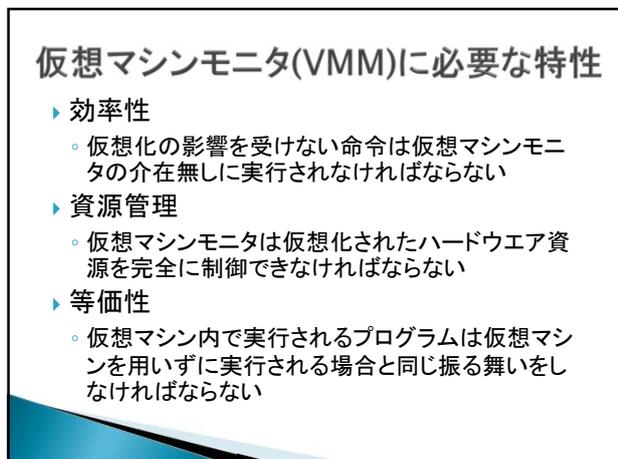
2



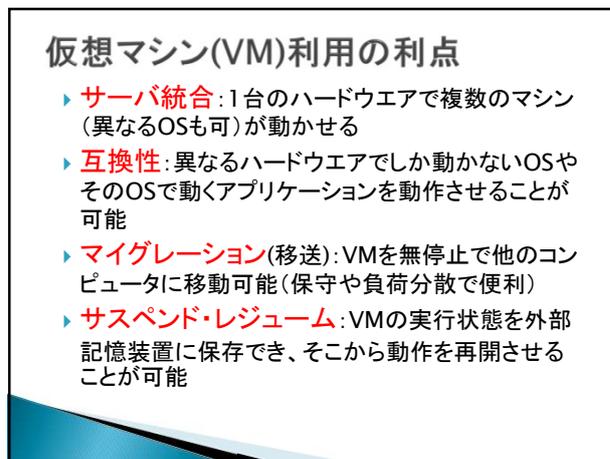
3



4



5



6

### 仮想プロセッサ(教科書11.2)

▶ 仮想プロセッサ ⇔ 実プロセッサ(物理プロセッサ)

仮想マシン1号プロセッサ 仮想マシン2号プロセッサ VMMのスケジューラが仮想プロセッサを実プロセッサに割当てて

仮想マシンモニタ(VMM) 実プロセッサ

**CPUに対するプロセス割当てとほぼ同じ動作**

7

### 仮想プロセッサの状態遷移

開始 → レディ状態(実プロセッサ待ち状態) = いつでも実行可能 → 選択 → 実行中状態 = 実プロセッサを割当てられ実行中 → 終了

他の仮想プロセッサへ譲る

待ち状態(事象待ち状態) = 入出力完了などの事象(イベント)を待つ

Sleep

Wakeup

割込みに対する制御は普通のOSと異なり「**エミュレート**」が必要

8

### エミュレーション(1)

▶ 特権命令をエミュレートする必要性

- 非特権モードで特権命令を実施すると内部割込みが発生、その時?
- 通常のOSでは
  - ・カーネルが制御
- VMでは
  - ・VMMが制御

▶ エミュレーションとは

- 割込み発生時にVMMが割込み原因を調べ、当該命令を**疑似実行**すること

9

### エミュレーション(2)

▶ センシティブ命令をエミュレートする必要性

- センシティブ命令: システム状態を変更したり、システムの状態によって動作が変わる命令
- これらは本来特権命令だが、プロセッサにより特権でないもの(= **内部割込み非発生**)も有る ∴ 特権命令と異なる制御が必要

▶ エミュレーション手法

1. 「バイナリ変換」によりカーネル内のセンシティブ命令を(内部割込みを起こす)特殊命令に書き換える
2. 準仮想化OSを活用(ハイパーコール)

10

### プロセッサの仮想化支援

▶ 仮想化支援機構としてOSにモードを追加

- ホストモード: VMMを動作させるためのモード
  - ・ホストモードの特権モード: VMMが動作
  - ・ホストモードの非特権モード: 必要に応じVMMを補助
- ゲストモード: VMを動作させるためのモード
  - ・ゲストモードの特権モード: OSのカーネルが動作
  - ・ゲストモードの非特権モード: 応用プログラムが動作

|        | ホストモード | ゲストモード  |
|--------|--------|---------|
| 非特権モード |        | 応用プログラム |
| 特権モード  | VMM    | OS      |

11

### メモリ仮想化(教科書11.3)

▶ 仮想マシンのメモリ = **ゲストメモリ**

- OSの仮想メモリと区別するため

仮想マシン(VM)1号 仮想マシン(VM)2号

ゲスト仮想メモリ ゲスト仮想メモリ

ゲスト物理メモリ ゲスト物理メモリ

仮想マシンモニタ(VMM)

メモリ

VM内の仮想メモリ

VMが提供する疑似的な実メモリ

**ホスト物理アドレスの計算方法は?**

12

### VMにおけるページング

1. シャドウページング (教科書p.159)
  - VMM内に「シャドウページテーブル」と呼ばれるアドレス変換表を作る
2. 準仮想化ページング (教科書p.160)
  - VM内に「準仮想化ページテーブル」を作る
3. ネステッドページング (教科書p.161)
  - 「ゲストページテーブル」と「ホストページテーブル」の2つを用いる(変換が2回以上必要)
  - ハードウェアの仮想化支援機構を用いる

13

### 入出力仮想化(教科書11.4)

▶ 仮想デバイス vs 実デバイス(物理デバイス)

入出力では必ず割り込みが発生  
→ VMMによる制御が必要

14

### デバイスエミュレーション

▶ 「デバイスエミュレータ」を用いる2つの方法

1. VMMが行うエミュレート
2. VMが行うエミュレート

15

### 準仮想化ドライバ

▶ 仮想デバイス専用につられたドライバ

他に「デバイスパススルー」もあり(教科書p.165)

16

## 第12章 ネットワークの制御

花田 英輔  
(このPowerPointは渡辺名誉教授作成のものを花田が一部改題した)

ネットワーク制御については「情報ネットワーク」で詳しく!

17

### OSにおけるネットワーク機能の位置付け(教科書12.1)

1. 入出力としてのネットワーク
  - 通信制御機構・通信回線を、個々の通信手順に従って使用
2. 論理的通信路としてのネットワーク
  - ファイルアクセス類似の論理的通信路=ソケット機能
3. 共有資源としてのネットワーク
  - ファイルサーバやプリントサーバの提供
4. プログラムが存在する場所としてのネットワーク
  - 遠隔手続き呼出し(RPC、Remote Procedure Call)、分散アプリケーション
5. ネットワークを見えなくする
  - ネットワーク全体で一つのシステム、分散OS

4まで実現、5は発展途上

18

### 通信インタフェース:プロトコル(教科書12.2)

▶ **プロトコル(Protocol)**

- 規定、議定書、儀典:外交の場や国際的催しで、その実務や交流の場における公式な規則や手順などを、ひとつの典拠として利用できるよまとめたもの

▶ **通信プロトコル**=通信規約、通信機器間で受け渡されるデータの形式と意味を定義

- これが一致しないと通信できない⇒**標準化**必要
- 通信技術は著しい進歩⇒**拡張性**必要

↓  
両立

▶ **ネットワークアーキテクチャ**=プロトコルの体系化

- 多くのレベルの手続きを階層化
- ⇒ **拡張性と標準化を両立**、処理を簡単化

19

### インターネットの4階層モデル

構築当初の概念

|                               |   |  |
|-------------------------------|---|--|
| アプリケーション層                     | HTTP,POP,FTP,Telnet,,                           | 応用目的に合わせた制御<br>(多様な目的で次々新方式)               |
| トランスポート層                      | TCP,UDP   | 送信・受信端間での<br>データフローの制御                     |
| ネットワーク層                       | IP,ICMP,IGMP                                    | ネットワーク上での<br>パケット配送制御                      |
| リンク層<br>(ネットワーク・インタ<br>フェース層) | Ethernet,ADSL,WiFi,<br>(デバイスドライバ<br>インタフェースカード) | 直接接続された機器間での<br>データ送受を制御<br>(電子技術進歩で次々新方式) |

【共通化】

•郵便システム

- 輸送手段(下位層)と内容物(上位層)は時代と共に変化
- 住所・切手を付けて出すと収集・配布(中間層)は不変

20

### アプリケーション層プロトコル例

- ▶ DHCP: Dynamic Host Configuration Protocol
- ▶ DNS: Domain Name System
- ▶ FTP: File Transfer Protocol
- ▶ HTTP: **Hypertext Transfer Protocol**
- ▶ IMAP: Internet Message Access Protocol
- ▶ LDAP: Lightweight Directory Access Protocol
- ▶ POP: **Post Office Protocol**
- ▶ SMTP: Simple Mail Transfer Protocol
- ▶ SNMP: Simple Network Management Protocol
- ▶ SSH: Secure Shell
- ▶ Telnet: Teletype network
- ▶ ...

多様な応用に対応して多様な約束事

21

### OSI参照モデル

▶ **OSI(Open System Interconnection) Reference Model**

- 国際標準化機構(ISO, International Organization for Standardization)が策定。**7階層**。
- ネットワークの基本モデルとして広く参照。

|            |                     |
|------------|---------------------|
| アプリケーション層  | 具体的通信サービス(メール等)     |
| プレゼンテーション層 | データ表現法(コード変換等)      |
| セッション層     | 通信開始~終了の制御(接続断の回復等) |
| トランスポート層   | 送信端と受信端の間の通信管理(再送等) |
| ネットワーク層    | 通信経路選択、データの中継配送     |
| データリンク層    | 直接接続された機器の間のデータ授受   |
| 物理層        | 物理的な接続(コネクタ形状、電圧等)  |

22

### インターネットの5階層モデル

現段階での概念=OSIモデルに準拠して語られる

| ISO参照モデル   | インターネット   |                 |
|------------|-----------|-----------------|
| アプリケーション層  | アプリケーション層 | 上位層の分割<br>はあいまい |
| プレゼンテーション層 |           |                 |
| セッション層     | トランスポート層  | L4 (Layer4)     |
| トランスポート層   |           |                 |
| トランスポート層   |           |                 |
| ネットワーク層    | ネットワーク層   | L3 (Layer3)     |
| データリンク層    | データリンク層   | L2 (Layer2)     |
| 物理層        | 物理層       |                 |

23

### TCP,UDP,IPの位置付け

送信ホスト ← 受信ホスト

送信ホスト L5 App ← 応用プロトコル → App L5 受信ホスト

送信ホスト L4 TCP/UDP ← トランスポートサービス インタフェース TCP/UDP → L4 受信ホスト

送信ホスト L3 IP ← IP → L3 受信ホスト

送信ホスト L2 Ethernet ← → L2 受信ホスト

送信ホスト L1 UTPケーブル ← 光ケーブル → L1 受信ホスト

ルータ

どこまでOSがサービス? ⇒ UNIX-BSD版でL4まで

24

### TCPとUDP

- ▶ **TCP (Transmission Control Protocol)**
  - コネクション型通信
  - 論理的通信路 **オープン⇒データ送受⇒クローズ**
  - その間にエラー、未達、順序制御等の処理を含む
  - 信頼性の高い通信
  - 一般的アプリケーションはこちらを使用
- ▶ **UDP (User Datagram Protocol)**
  - コネクションレス型通信
  - 単にデータ送付
  - オープン・クローズやエラー、未達、順序制御等の機能なし
  - 信頼性より即時性が必要な用途 (画像・音声のストリーミング、DNS、..)

25

### 通信用プログラミングインタフェース (教科書12.3)

- ▶ **ソケット**=インターネット通信のためのAPI
  - UNIXのBSD版で開発、Windows等もサポート

26

### ソケット通信処理の流れ(APIの使用)

27

### クライアント・サーバ方式(教科書12.4)

- ▶ サービス提供側とサービス要求側を役割分担

クライアントプログラム (多数プロセス、作業が終われば終了)

サーバプログラム (1プロセス、要求を待って無限ループ) 多数クライアントを並行に処理の工夫 => 子プロセス(スレッド)を作成、各々の相手

28

### ピアツーピア通信

- ▶ **Peer-to-Peer** (Peer=同僚・中間・対等者)
- ▶ サービスの提供側と要求側の区別なく対等な通信

29

### 遠隔手続き呼出し

- ▶ **RPC (Remote Procedure Call)**
  - ネットワークの先の手続き/関数を呼び出すこと
  - クライアント・サーバ方式の拡張

30

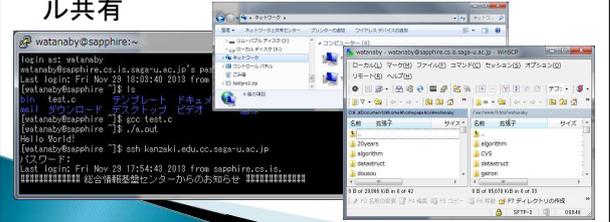
### 分散オブジェクト技術

- ▶ オブジェクト指向=システムをオブジェクトで構成、  
それらがメッセージ交換  
⇒ネットワーク環境に適合
- ▶ 様々な技術が発展
  - CORBA、DCOM、RMI、..
- ▶ 遠隔地のオブジェクト同士でメソッド呼出し  
=RPCと同様に実現

31

### ネットワークを介したOS機能の利用 (教科書12.5)

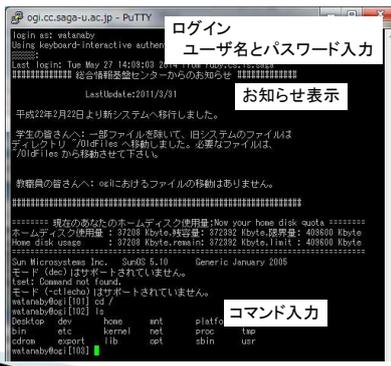
- ▶ 遠隔利用: telnet, ssh(secure shell)
- ▶ ファイル転送: ftp(file transfer protocol)、scp(secure copy)
- ▶ ファイル共有: UNIXのNFS、Windowsのファイル共有



32

### SSHの使用

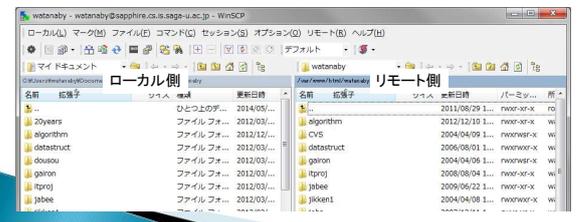
- ▶ 遠隔のコンピュータにコマンド入力・実行
- ▶ コマンド「ssh xxx.jp」
- ▶ Windows用 PuTTY、TeraTerm、..



33

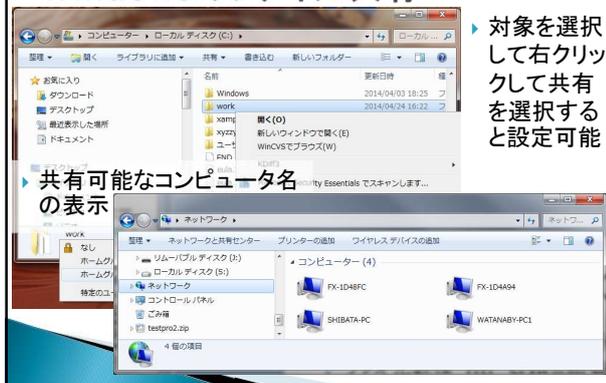
### SCPの利用

- ▶ 遠隔のコンピュータとの間でファイルコピー
- ▶ コマンド入力で利用
  - 例: `scp -r hana@xxx.jp:temp temp`  
xxx.jpホストのユーザーhanaのホームディレクトリのtemp以下を、再帰的に(-r)、ローカルのtemp以下にコピー
- ▶ GUIソフトで利用(下はWindows用のWinSCPの例)



34

### Windowsのファイル共有



35

### 今回の課題

1. 仮想プロセッサのスケジューリングアルゴリズムについて調べ、それぞれの利欠点を記せ
  2. クライアントサーバ方式のシステムにおけるクライアントとサーバの役割を述べよ
  3. (予習)「セキュリティの三要素」について調べ、その内容を説明せよ
- ▶ 今回のファイル名は「学籍番号-OS13.docx」(例: 24238000-OS13.docx)としてください
  - ▶ 締切: 1月23日(金) 18:00 (遅れた場合は減点)
  - ▶ 本講義に関する情報は(この講義資料も)次のWebpageにも掲載するので、時々参照すること  
<https://www.ai.is.saga-u.ac.jp/~hanada/OS/>

36