

復興知プロジェクト ロボット講習会 in 南相馬

中村啓太

復興支援センター

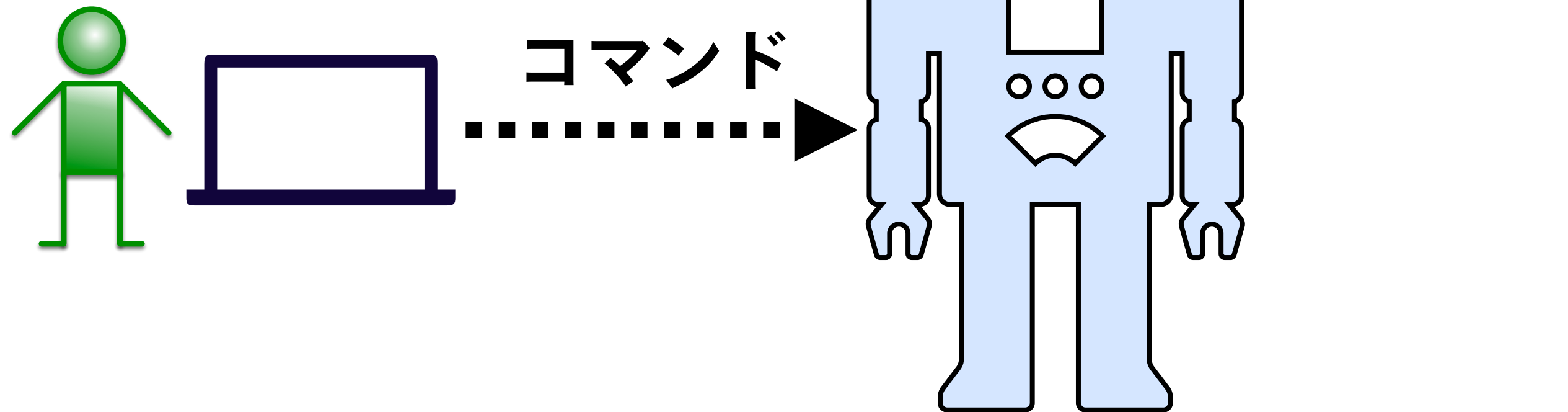
2018/11/10

発表構成

- ❖ コンピュータ（計算機）について
- ❖ 使用機材
- ❖ 遠隔操作の基礎知識
- ❖ Linux OSの使用方法
- ❖ OpenRTMの概要
- ❖ OpenRTMの開発環境構築
- ❖ 本日の課題

本講習で行うこと

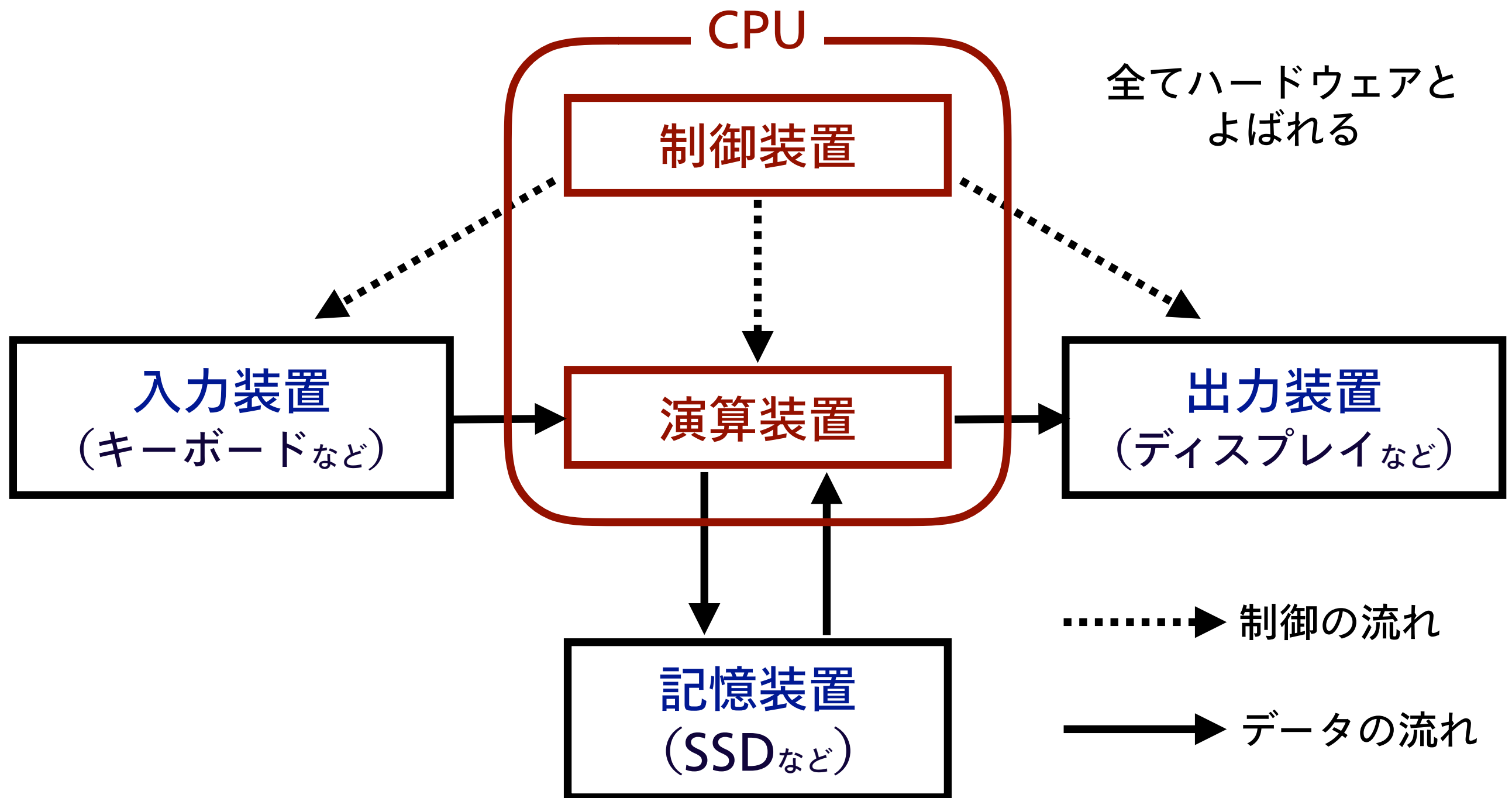
❖ コンピュータでロボットを動かす



コンピュータ(計算機)について

コンピュータ(計算機)とは？

❖ コンピュータの五大装置



例：ノートパソコン

- ❖ 入力装置：キーボード，マウス，トラックパッド
- ❖ 出力装置：ディスプレイ，プリンタ
- ❖ 記憶装置：SSD，ハードディスク，USBメモリ
- ❖ 制御装置 + 演算装置：CPU



OS(オペレーティングシステム)とは？

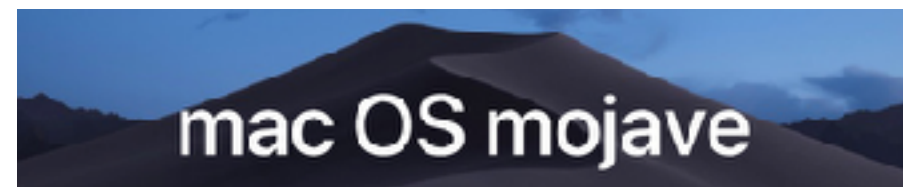
❖ コンピュータの五大装置をどのように扱うかのための
システムプログラムを集めたもの（ソフトウェア）

❖ 代表的なOS

❖ Windows



❖ macOS



❖ **Linux**：本講習で扱う

❖ iOS



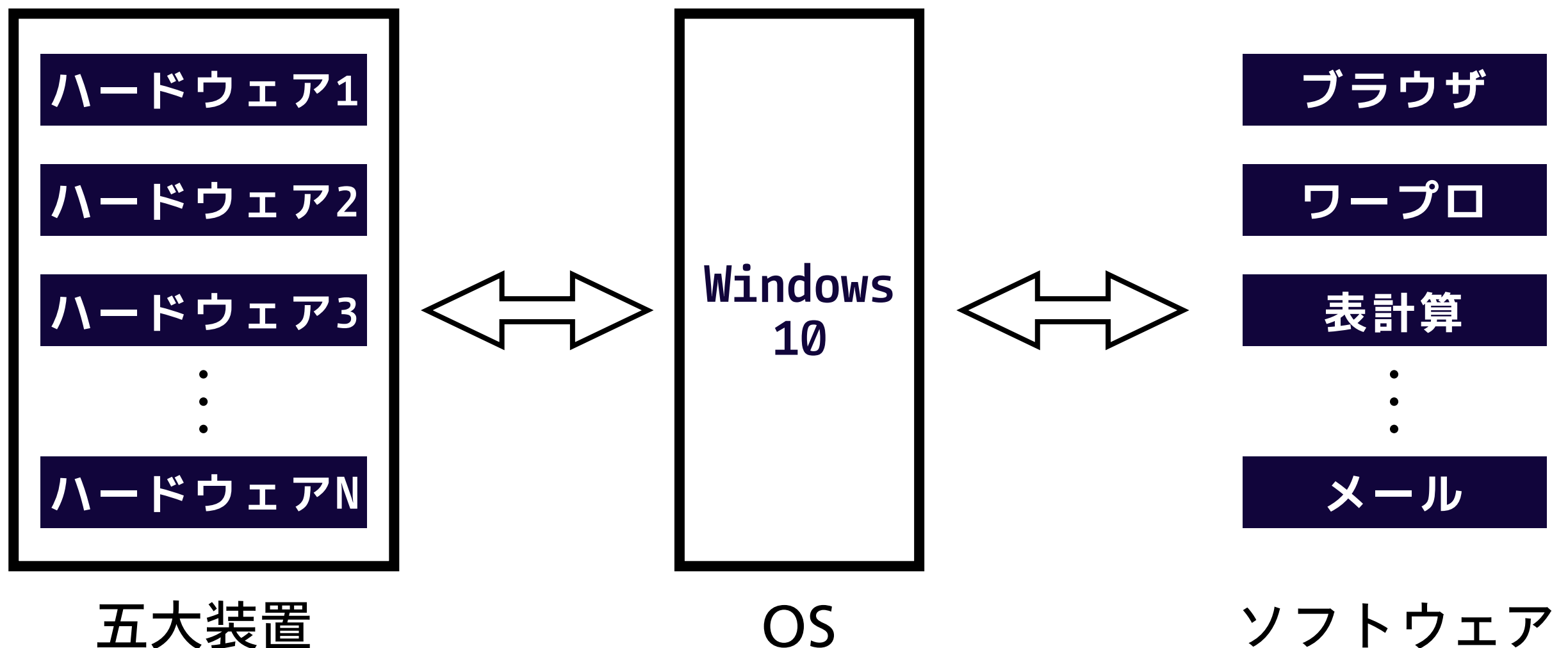
❖ Android



OS(オペレーティングシステム)とは？

- ❖ OS（オペレーティングシステム）はハードウェアとアプリケーションを仲介するソフトウェアである

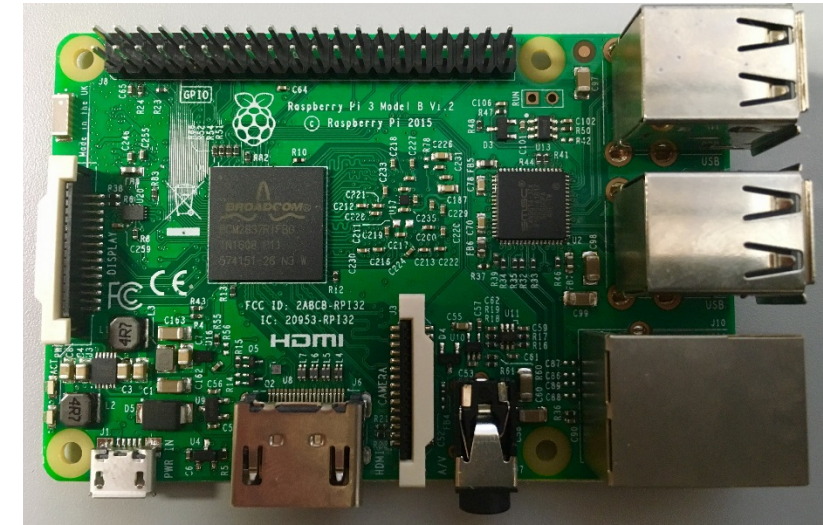
例：



機材説明

Raspberry Pi

- ❖ CPUがついた超小型コンピュータ
 - ❖ 安価 (6000円弱) かつ消費電力が低い
 - ❖ microSDカード, マウス, キーボード, ディスプレイなどを接続できる
 - ❖ microSDカードにOSを書き込めば利用可能
- ❖ ハードウェアに繋げやすいため, 電子工作が容易にできる
- ❖ 『プログラミング』 + 『電子工作』 + 『インターネット』 で様々なことができる



LEGO MINDSTORMS EV3

- ❖ マサチューセッツ工科大学と共同開発されたロボティクス製品
 - ❖ 教育用二輪移動ロボット
 - ❖ ジャイロ，カラー，タッチセンサなど多くのセンサを持つ
 - ❖ Java，C++，Pythonなど多くの言語で動かすことが可能



使用OS

- ❖ 講習会ではノートPC, Raspberry PiとEV3を使用
- ❖ Raspberry PiとEV3のOSはWindowsではなく **Linux**を使用



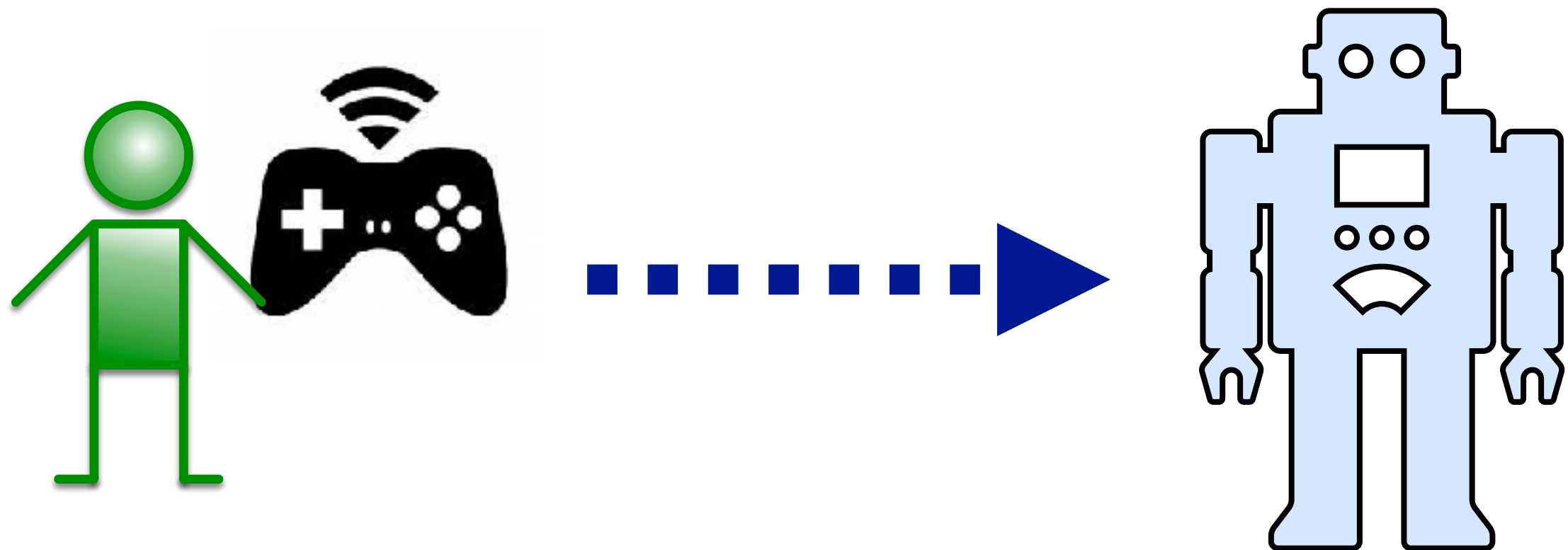
Raspbian
(Linux)



EV3DEV
(Linux)

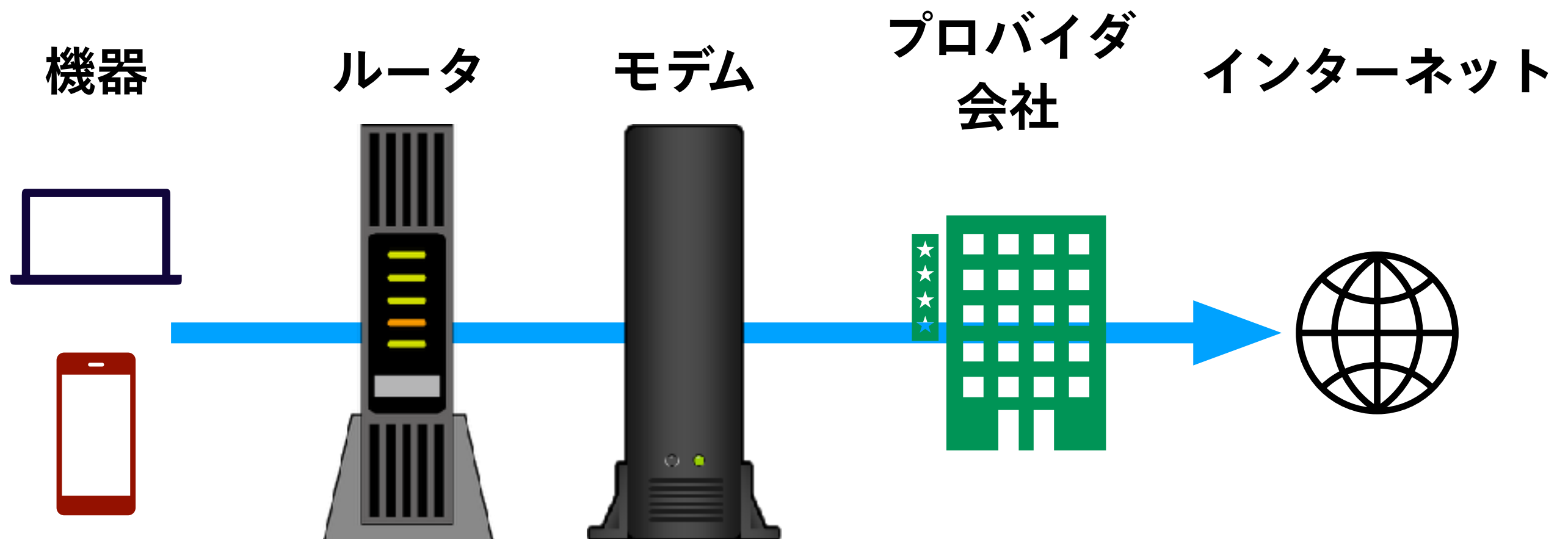
遠隔操作の基礎知識

ロボットの遠隔操作



通信さえ確保できていれば， リモコンなどで操作が可能

インターネット接続



※ルータとモデムが一緒になっていることもある

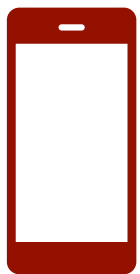
IPアドレスの割り当て

同一ネットワーク環境下



機器A

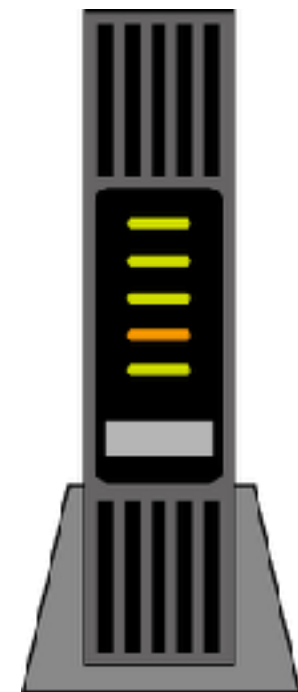
IPアドレス：192.168.2.1



機器B

IPアドレス：192.168.2.2

ルータ



ルータに繋がった際に、各機器にIPアドレスが与えられる

コンピュータのリモートログイン

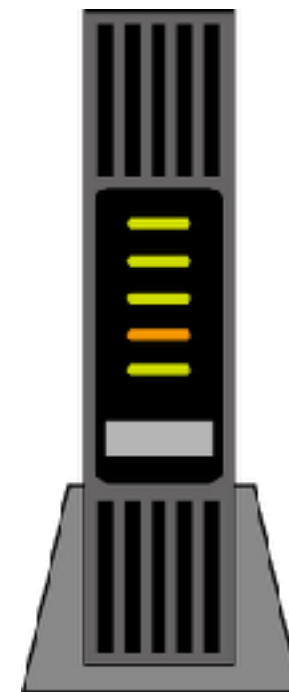
同一ネットワーク環境下

機器A

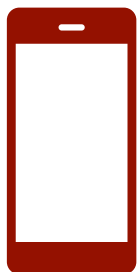


IPアドレス : 192.168.2.1

ルータ



機器B



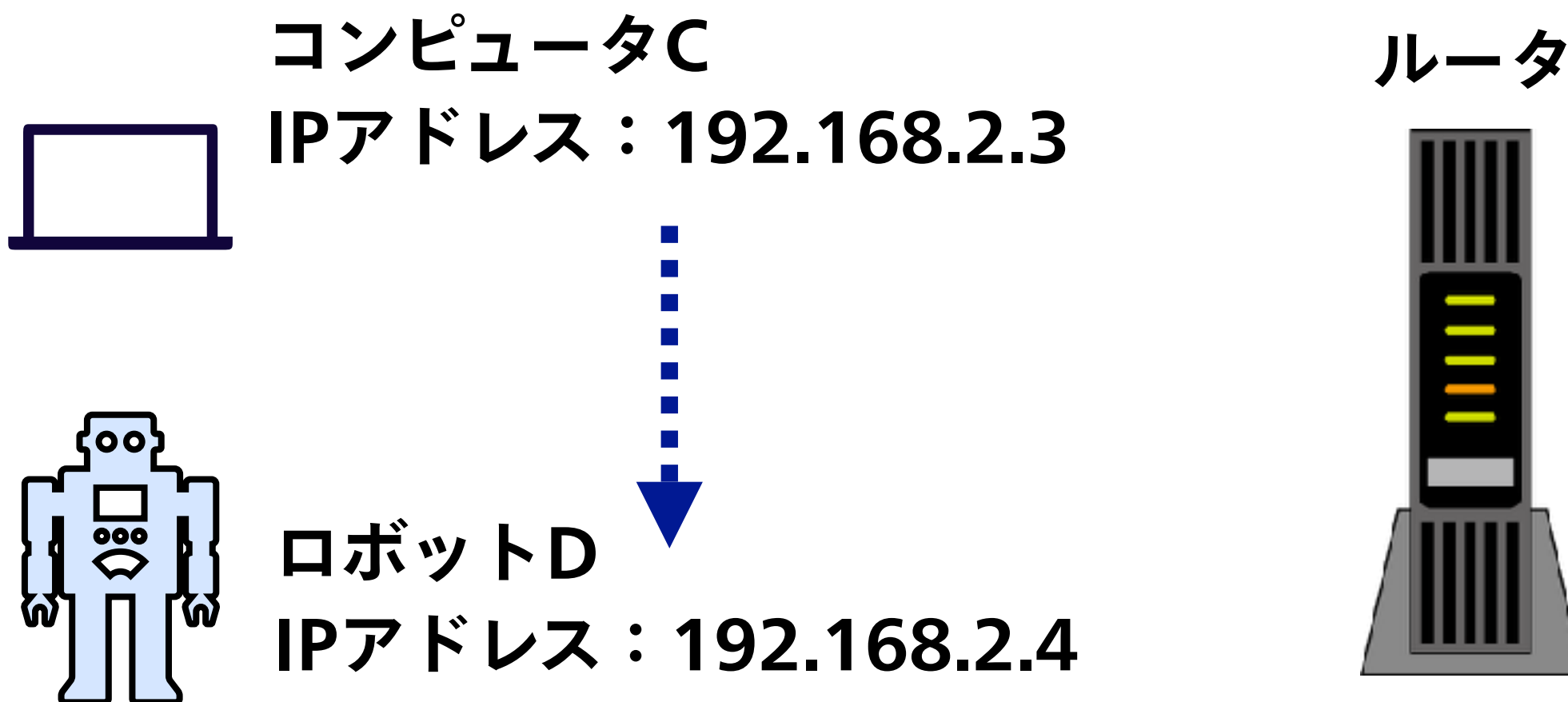
IPアドレス : 192.168.2.2



IPアドレスを使用して機器Aから機器Bに遠隔操作できる

ネットワークによる遠隔操作

同一ネットワーク環境下



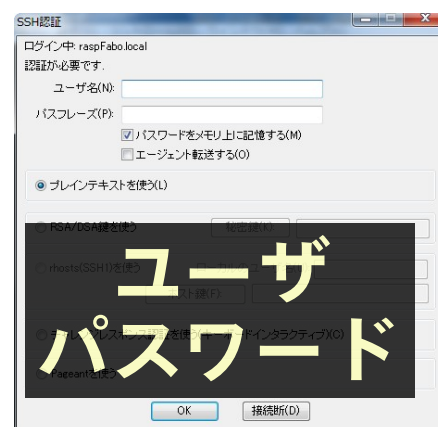
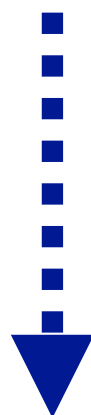
同一ネットワーク環境下にあるコンピュータから、
ロボットに命令を出すことができる

本講習会での遠隔操作

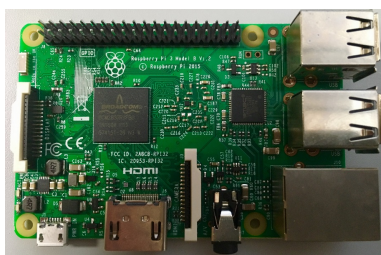
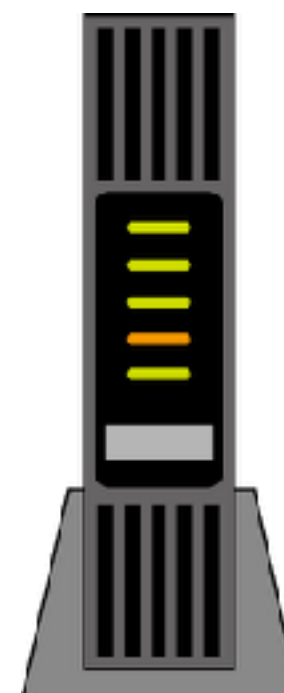
同一ネットワーク環境下

ノートパソコン

IPアドレス：192.168.2.9



ルータ



RaspberryPi

IPアドレス：192.168.2.10

TeraTermを使用してRaspberryPiにアクセスする

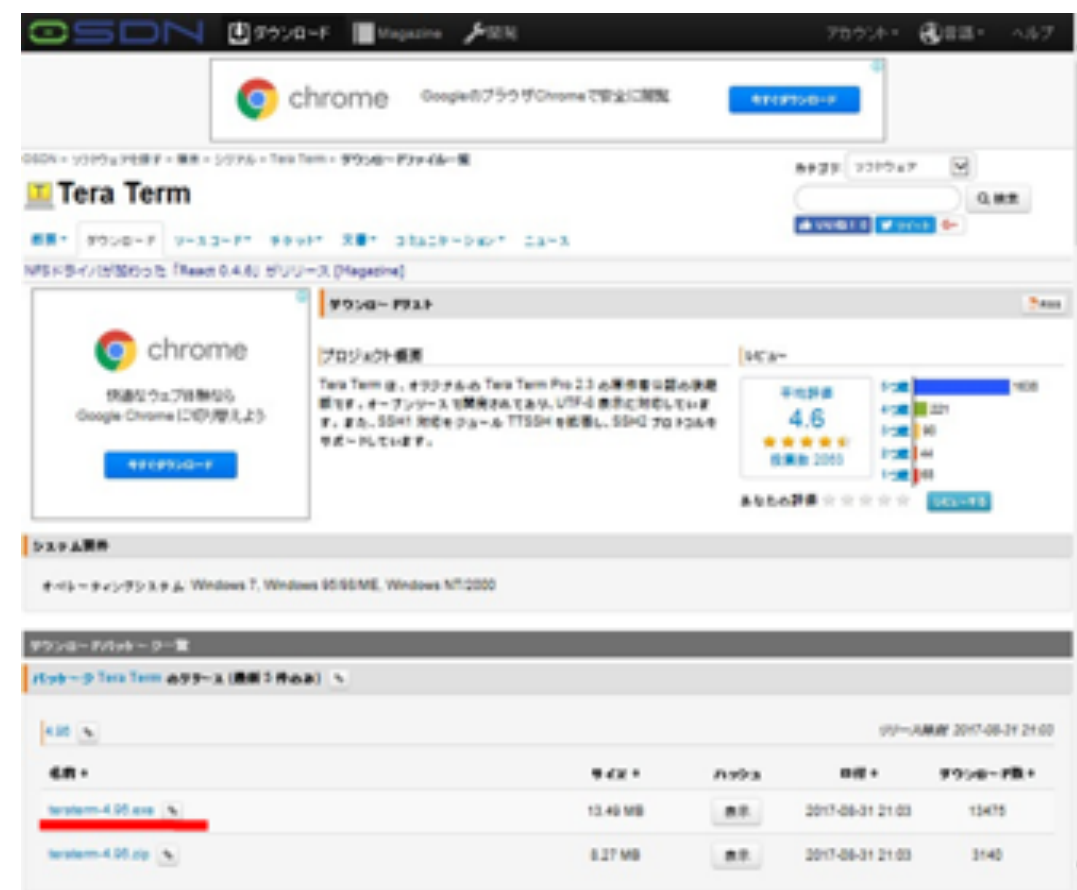
TeraTermをインストール

1. 以下URLへアクセス

<https://ja.osdn.net/projects/ttssh2/releases/>

2. TeraTermをダウンロード teraterm-X.XX.exeをダウンロード (最新をダウンロード)

3. TeraTermをインストール teraterm-X.XX.exeを ダブルクリックしてインストール



Bonjourインストール

1. iTunesインストールで、Bonjourがインストールされる
<https://www.apple.com/jp/itunes/download/>

最新のiTunesは、
macOS Mojaveに含まれています。

最新バージョンのiTunesはmacOS Mojaveと一緒にインストールされます。
今すぐアップグレードして、あなたのお気に入りの音楽、映画、Podcastを楽しみましょう。
iTunesではApple Musicのメンバーにもなれるので、5,000万曲以上をストリーミングしたり、
ダウンロードしてオフラインで再生できるようになります。広告は一切ありません。

 Windows用のiTunes

今すぐダウンロード(64ビット版)

**Bonjourを使用すると
ホスト名でログインが可能**

Windows 32ビット版をお探しですか? [ここからダウンロード](#) ↓

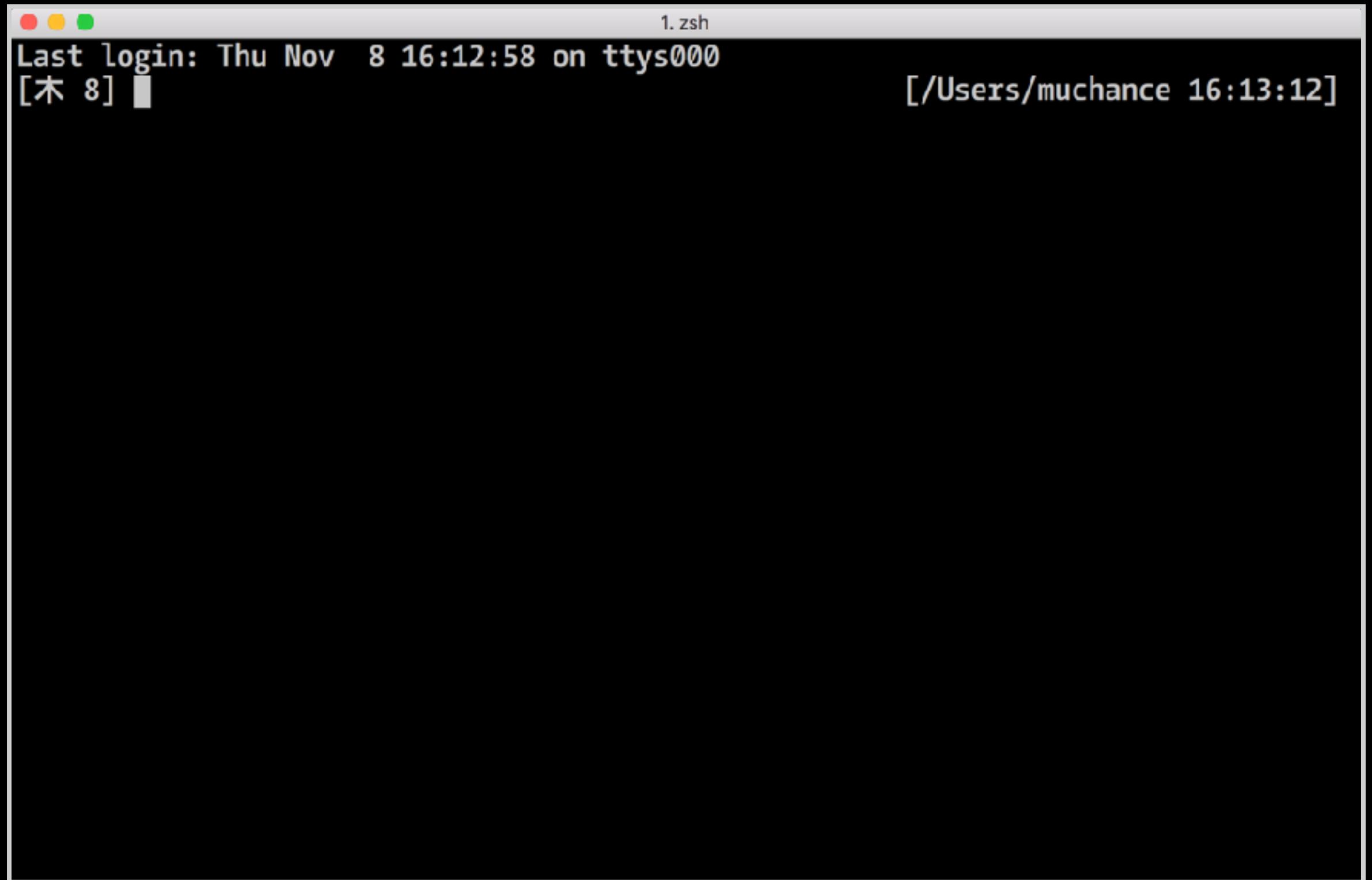
Linux OSの使用方法

Linuxとのやりとり

1. GUI (Graphical User Interface) :
マウスを使ってアイコンなどの視覚的な要素を操作する方式
 2. CUI (Character-based User Interface) :
キーボードを使って文字情報だけで操作する方式
- ❖ Linuxでは基本的にCUIでやりとりを行う
 - ❖ 慣れるとGUIよりも素早く操作ができる
 - ❖ 一連の作業を自動化しやすい
 - ❖ コンピュータをネットワーク経由で遠隔操作できる



GUIの例



A terminal window titled "1. zsh" with standard macOS window controls (red, yellow, green buttons). The terminal displays the following text:

```
Last login: Thu Nov  8 16:12:58 on ttys000
[木 8] █                                     [/Users/muchance 16:13:12]
```

The prompt "[木 8] █" indicates the current date and time. The path and time "[/Users/muchance 16:13:12]" are shown on the right side of the terminal.

CUIの例

ターミナルとコマンド

- ❖ 文字入力でコンピュータとやりとりする場合、ターミナル(端末)やコンソールとよばれるソフトウェアを起動する
- ❖ ターミナルはコンピュータと文字でやりとりするため、対話型インターフェースとよばれる
- ❖ 本当に自由にやりとりするわけではなく、あらかじめ決められたプログラムを使い命令をする
 - ❖ この命令のことをコマンドとよぶ
- ❖ ターミナルは、コマンドに対して結果を文字列で表示する

ログインとログアウト

- ❖ OSを利用するにはログイン（ログオン）が必要
- ❖ OSを利用できるのは、システムにあらかじめアカウントが登録されたユーザーに限定される
- ❖ ユーザ名とパスワードの組み合わせで認証を行う
- ❖ ログインを終えた場合や端末を起動した場合、プロンプトが表示される
 - ❖ コマンドを受け付ける準備ができていることを表す
- ❖ OSの利用を終えたらログアウト（ログオフ）する
 - ❖ **exit** とコマンドを書くことで、ログアウトができる



ユーザー名

パスワード



[パスワードのリセット...](#)



Windows7 ログイン画面

http://img-cdn.jg.jugem.jp/f04/32604/20110422_2054022.png

ユーザーの切り替え(W)

ログオフ(L)

ロック(O)

再起動(R)

スリープ(S)

休止状態(H)

シャットダウン



管理者ユーザ

- ❖ OSには、システム変更の権限を持った管理者アカウントが存在
 - ❖ Windowsの場合：Administrator
 - ❖ Linuxの場合：root
- ❖ 通常のアカウントを一般ユーザ、管理者をスーパーユーザとよぶ
- ❖ インストールなどのOSに変更を加える処理を行う場合、一般ユーザに管理者権限を持たせてコマンドを行う

sudo apt update など

コマンド操作

- ❖ コマンドを入力し、「Enter」キーを押すとコマンドが実行
- ❖ コマンドに引数やオプションをつけることができる
 - ❖ 引数やオプションをつけることで細かくコマンドを行える
 - ❖ **#**の右側はコメント（説明文）であり、コマンドではない
- ❖ 例：**cal**コマンド：カレンダーを表示する（calendar）
 - cal** # 現在の年月のカレンダーを表示
 - cal 2018** # 2018年のカレンダーを表示（2018が引数）
 - cal 11 2018** # 2018年11月のカレンダーを表示（11と2018が引数）
 - cal -3** # 先月と翌月のカレンダーを表示（-3がオプション）

コマンドの調べ方

- ❖ Linuxではオンラインマニュアルが用意されている
- ❖ マニュアルは**man**コマンドで参照可能(manual)
- ❖ コマンドの使い方
 - ❖ 引数やオプションの指定方法
 - ❖ オプションの意味コマンドの出力する内容
 - ❖ 「q」キーを押すと、マニュアル参照を止める
- ❖ 使用例：**cal**コマンドのマニュアルを表示する場合

man cal

システムの終了と再起動

- ❖ 管理者権限を持つことでコマンドからシステムの終了や再起動を行うことができる(**sudo**コマンドを使用)

sudo shutdown -h now # ただちに終了する

sudo shutdown -r now # ただちに再起動する

sudo shutdown -r +10 # 10分後に再起動する

sudo shutdown -r 21:00 # 21時に再起動する

ユーザーの切り替え(W)

ログオフ(L)

ロック(O)

再起動(R)

スリープ(S)

休止状態(H)

シャットダウン



Windows7 システムの終了と再起動画面

ファイルの命名規則

- ❖ Linuxでは、テキスト・画像・設定などのデータは、すべてファイルという単位で扱われており名前がついている
- ❖ 日本語を扱うことも可能だが、使うソフトウェアによっては正しく認識されないこともあるため、基本的には以下のような文字だけをファイル名として用いる
- ❖ 半角アルファベット，半角数字，
- ❖ 半角アンダーバー(`_`)，半角ハイフン(`-`)，半角ドット(`.`)
- ❖ 基本的に半角文字しか認識できない（全角文字はエラーの原因）

ファイルの拡張子

- ❖ Linuxには『.txt』, 『.docx』のような拡張子の概念はない
- ❖ ファイルを扱うアプリケーションは拡張子による判別が多く、ユーザがどのようなファイル形式かわかりやすくするために、一般的には拡張子を付けたほうがよい
- ❖ Linuxではドット『.』で始まるファイル名を隠しファイルとして扱われており、通常では表示されない
 - ❖ 隠しファイルは、ユーザの環境やアプリケーションの設定ファイルとして扱うことが多い

ファイルとディレクトリ

❖ 通常ファイル

- ❖ データやプログラムが格納されたファイルのこと
- ❖ テキスト情報が格納された人間の読むことができる
テキストファイルと、データやプログラムが格納され、
人間が見ると意味不明なバイナリファイルがある

❖ ディレクトリ

- ❖ Windowsのフォルダに相当するものをディレクトリとよぶ
- ❖ ディレクトリも特殊なファイルの形式として扱われる

ファイルとディレクトリ

❖ リンクファイル

- ❖ ファイル名とファイルの実体(内容)を紐付ける

- ❖ シンボリックリンクとハードリンクがある

❖ 特殊ファイル

- ❖ Linuxではデバイスを抽象化してファイルとして扱う

- ❖ 例：プリンタに対応するファイルにデータを書き込んだとき、プリンタから出力されるなど、ファイルを通してデバイスを扱えるようにする

ディレクトリ

❖ ファイルをまとめる入れ物のこと：

❖ Linux：ディレクトリ，Windows：フォルダ

❖ ディレクトリの中に，ディレクトリを作成することも可能

❖ つまり，ファイルやディレクトリは階層構造で管理される



親ディレクトリ



サブディレクトリ



ファイル

パス

- ❖ 階層構造で表現できるファイルやディレクトリの位置を表したものをパスとよぶ
- ❖ sampleディレクトリ内にtestというディレクトリがある場合, sampleディレクトリから見たtestディレクトリのことを **sample/test** のようにスラッシュ(/)で区切って表す

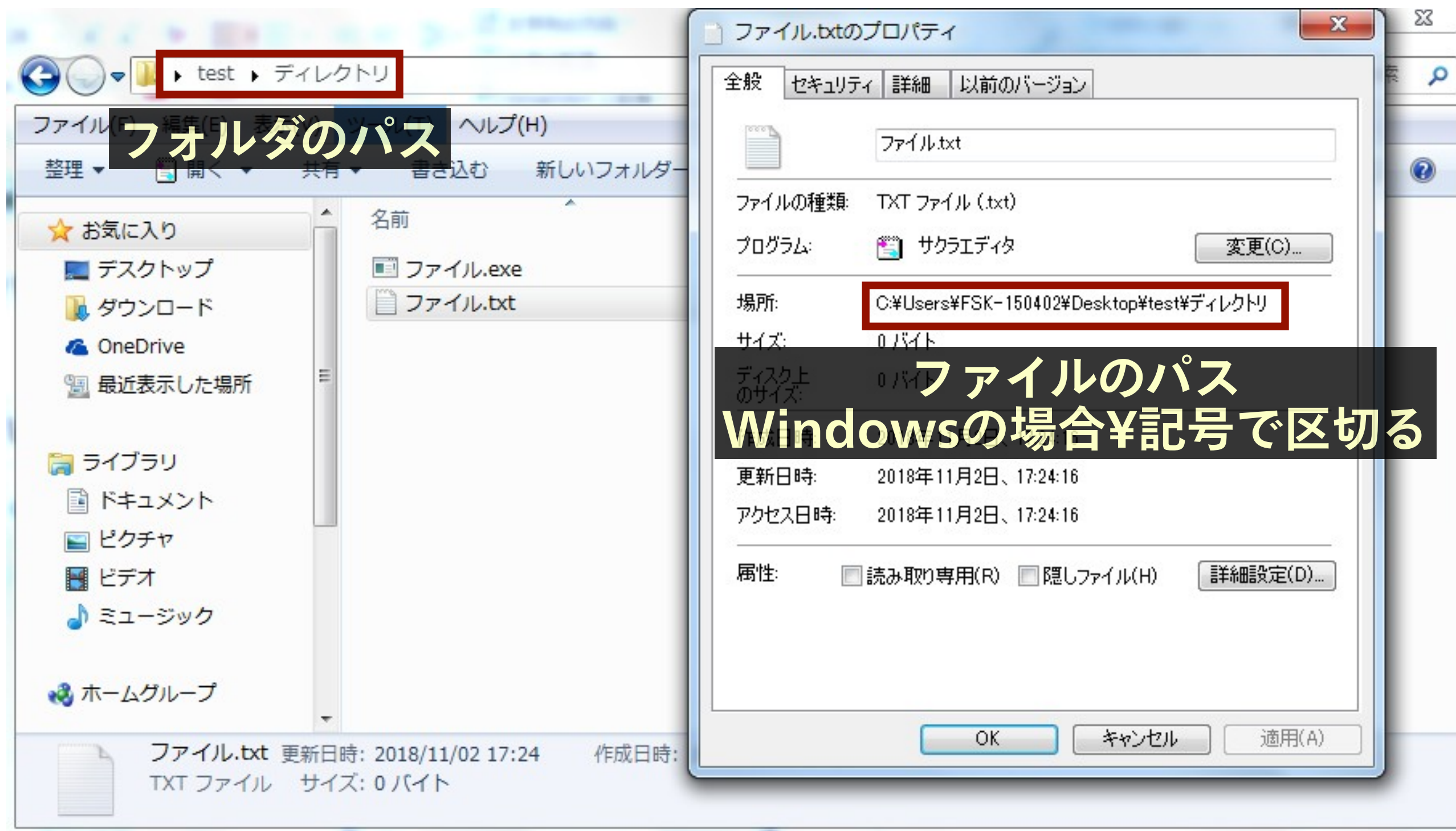


パス

- ❖ 階層構造で表現できるファイルやディレクトリの位置を表したものをパスとよぶ
- ❖ testディレクトリにmemo.txtというファイルがあれば、そのファイルは testディレクトリから見て**test/memo.txt**と表す

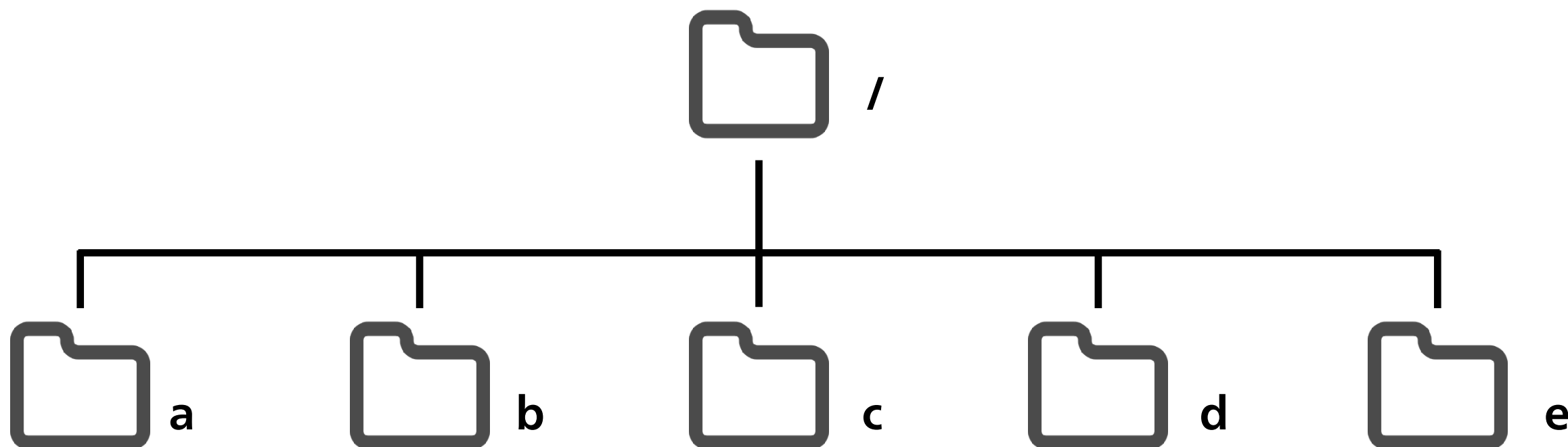


パス | Windowsの場合



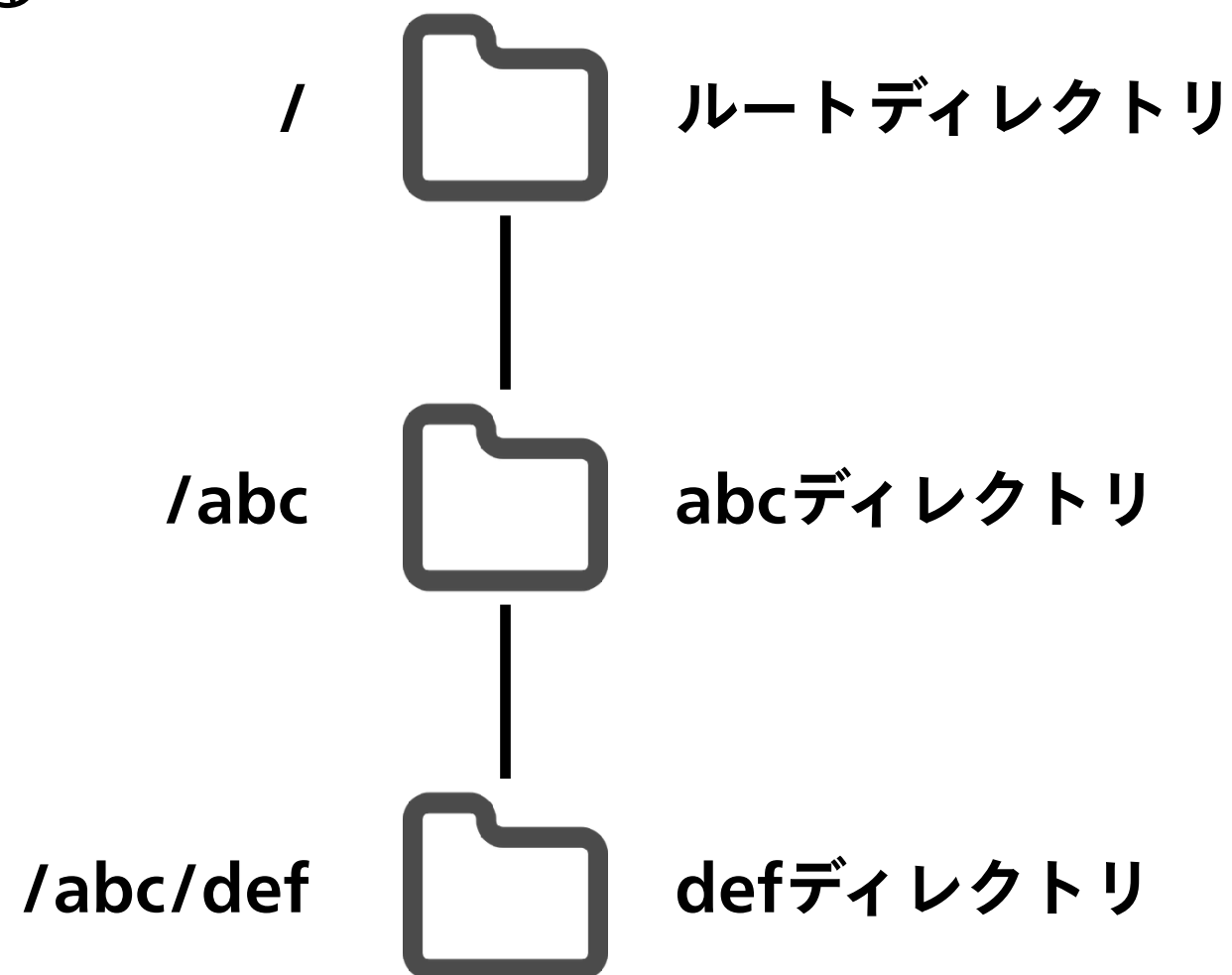
ルートディレクトリ

- ❖ Linuxではディレクトリは階層構造になっている
- ❖ その頂点となるディレクトリ、つまりすべてのディレクトリを格納しているトップディレクトリのことを
ルートディレクトリとよび、スラッシュ / 一文字で表す



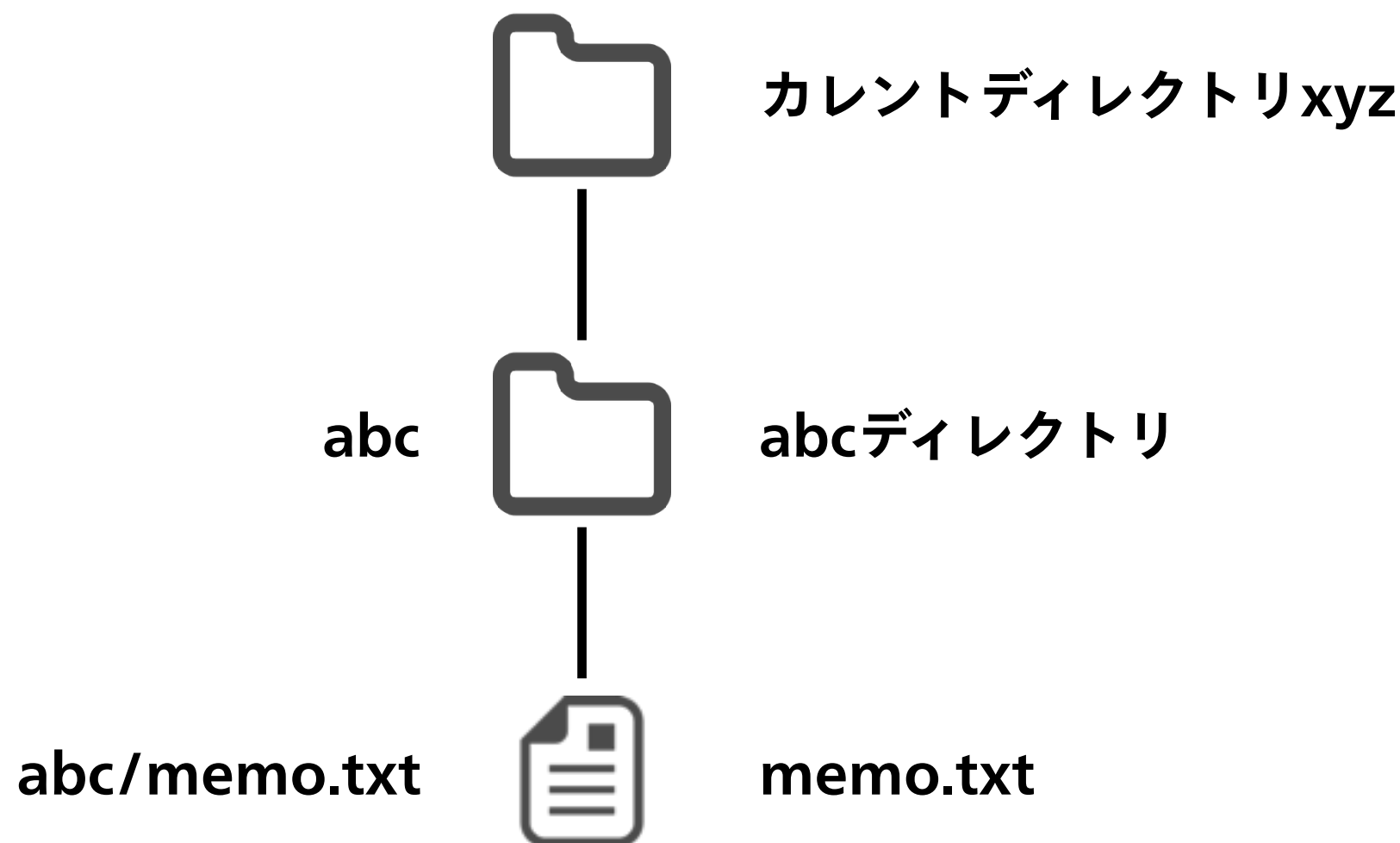
絶対パス

- ❖ ルートディレクトリを基点としてファイルやディレクトリの位置を表す
- ❖ ディレクトリツリー内のファイルやディレクトリを一意に特定できる



相対パス

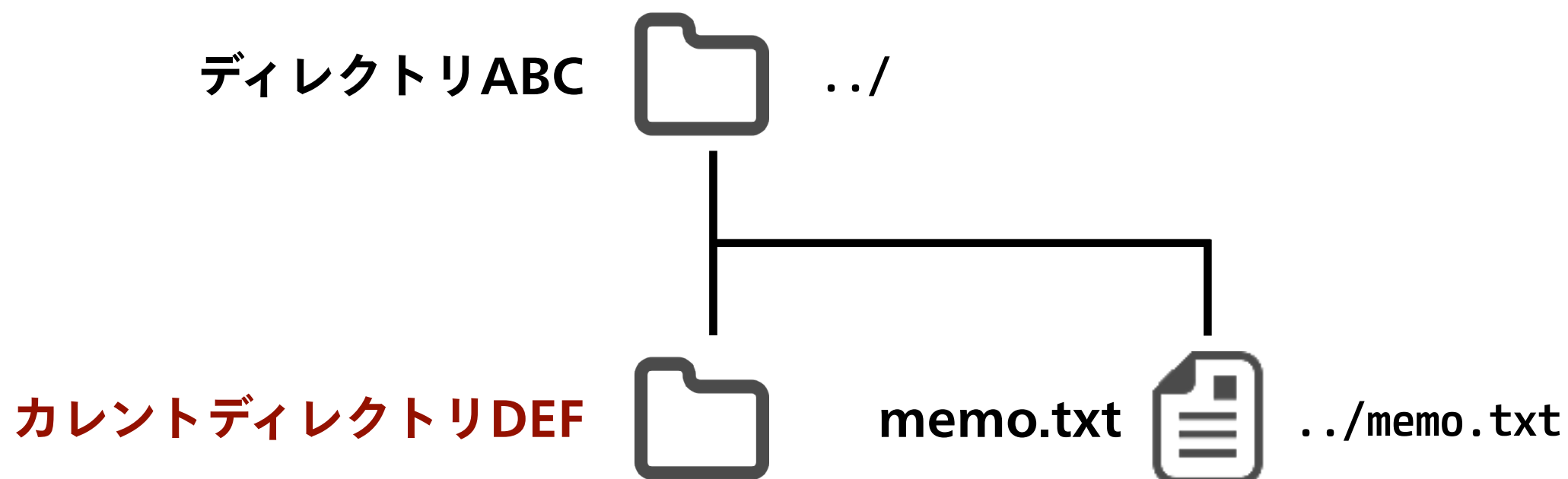
- ❖ ユーザが作業を行っているディレクトリ
（カレントディレクトリ）を基点としてファイルや
ディレクトリの位置を表す



相対パスで使用可能な記号

❖ 以下の記号を使用することで相対パスを表現できる

記号	説明
.	カレントディレクトリ
..	1つ上のディレクトリ
~(チルダ)	ホームディレクトリ



ホームディレクトリ

- ❖ 各ユーザごとに割り当てられるディレクトリ
 - ❖ 一般的に /home/ユーザ名 となる
- ❖ Linuxにログインすると、ユーザのカレントディレクトリはホームディレクトリになる

カレントディレクトリの移動

❖ 作業ディレクトリの移動はカレントディレクトリの変更と同義

❖ **cd** コマンド (change directory) : カレントディレクトリ移動

cd [ディレクトリパス]

cd # 引数が無い場合, ホームディレクトリに移動

❖ GUIだと, ディレクトリをクリックして開く作業に相当

カレントディレクトリパスの確認

❖ **pwd** (print working directory) コマンド :

ユーザが作業を行っているディレクトリパスを確認する

pwd # カレントディレクトリパスの確認

ディレクトリの作成/削除

- ❖ **mkdir**(make directory)コマンド：ディレクトリを作成する
mkdir [ディレクトリパス]
- ❖ **rmdir**(remove directory)コマンド：ディレクトリを削除する
rmdir [ディレクトリパス]
- ❖ 削除対象のディレクトリ内に1つでもファイルが存在する場合、ディレクトリの削除は行えない
- ❖ 削除したディレクトリの復元は行えない

ファイル一覧表示コマンド

❖ **ls**(list)コマンド：ディレクトリ内の全ファイル情報を表示

ls [ディレクトリパス]

ls # 引数がない場合、カレントディレクトリ内の一覧を表示

ls -l # オプション「-l」でファイルの詳細情報を表示

ls -a # オプション「-a」でドットファイルを含めて表示

ファイル内容確認と削除

❖ **cat**(concatenate)コマンド：ファイル内容を表示

cat [ファイルパス]

❖ **rm**(remove)コマンド：ファイルを削除

rm [ファイルパス]

ファイル/ディレクトリの移動

❖ **mv**(move)コマンド：ファイル/ディレクトリの移動

移動元ファイルを指定したディレクトリに移動する

mv [移動元ファイルパス] [移動先ディレクトリパス]

移動元ディレクトリを指定したディレクトリに移動する

mv [移動元ディレクトリパス] [移動先ディレクトリパス]

ファイル/ディレクトリの名前変更

❖ **mv**(move)コマンド：ファイル/ディレクトリの名前変更

ファイル/ディレクトリを指定した名前に変更する

mv [ファイル/ディレクトリパス] [変更する名前]

ファイル/ディレクトリのコピー

❖ **cp**(copy)コマンド：ファイルをコピー

コピー元ファイルを指定したファイル名で作成する

cp [コピー元ファイル名] [コピー先ファイル名]

❖ **cp** コマンドにオプション「**-r**」を付けることで、
ディレクトリのコピーを行える

コピー元ディレクトリを指定したディレクトリ名で作成する

cp -r [コピー元ディレクトリ名] [コピー先ディレクトリ名]

パーミッション

- ❖ 重要なファイルは、他ユーザが自由に閲覧したり、書き換えたりできないようにファイルを保護する必要がある
- ❖ システム上の重要なファイルを自由に書き換えられると、システム障害が発生する
- ❖ これらのトラブルが発生しないように、ファイルやディレクトリには、アクセス権が設定されている
- ❖ アクセス権は「読み込み」「書き込み」「実行」が存在する

パーミッションの確認

- ❖ パーミッションを設定することで以下のことができる
 - ❖ 自分はファイル内容の読み込み/書き込みを行える
 - ❖ 自分以外のユーザはファイル内容の読み込み/書き込みを行えない
- ❖ 各アクセス権は、以下の記号で表される
 - ❖ 読み取り：**r** (readable), 書き込み：**w** (writable)
実行：**x** (executable), 不可：**-** (ハイフン)
- ❖ ファイルのパーミッションは、**ls -l**コマンドで確認できる

ファイルの詳細情報

❖ **ls -l** コマンドではファイルの詳細情報を確認できる

❖ ファイルモード、リンク数、ファイルの所有者、
ファイルの所有グループ、ファイルのサイズ、
最終更新日時、ファイル名が表示されている

ls -l

drwxr-xr-x 3 d8161105 grdd 4096 11月 14 2014 App/

❖ ファイルモードの2文字目以降がパーミッションを表す

❖ 1文字目はファイルの種類を表す（dはディレクトリ）

パーミッションの見方

- ❖ 前ページの **rwXr-xr-x** の部分がパーミッションを表す
- ❖ 前から3文字ごとに各ユーザの権限を表している
 - ❖ 「所有者のアクセス権」 : **rwX**
 - ❖ 「所有者グループに属するユーザのアクセス権」 : **r-x**
 - ❖ 「その他のユーザのアクセス権」 : **r-x**
- ❖ 「-」 はアクセス権がないことを表す
- ❖ 「読み取り」, 「書き込み」, 「実行」 の順番

パーミッションの数値表記

- ❖ アクセス権を 読み取り：4，書き込み：2，実行：1 と表現可能
- ❖ 「所有者」，「所有者グループに属するユーザ」，「その他のユーザ」ごとに足し合わせた数値で表す
 - ❖ 各ユーザのアクセス権を各数値が表す
- ❖ パーミッション $rwXr-xr-x$ を数値表記した場合，755となる
 - ❖ 所有者のアクセス権： $rwX=4+2+1=7$
 - ❖ 所有者グループに属するユーザのアクセス権： $r-x=4+1=5$
 - ❖ その他のユーザのアクセス権： $r-x=4+1=5$

パーミッションの変更例

- ❖ `chmod(change mode)`コマンド：パーミッションを変更する
 - ❖ 管理者権限を持つことで使用可能
 - ❖ アクセス権には、755のように3桁の数値を指定する
- ❖ `diary.txt`のファイルのパーミッションを`rw-r-r--(644)`に設定
`chmod 644 diary.txt`
- ❖ `data`ディレクトリのパーミッションを`rwxr-xr-x(755)`に設定
`chmod 755 data`

プログラム生成

- ❖ **cmake** コマンド：makefileを作成する
- ❖ makefileとは、プログラム生成などを行うためのコマンドが記載されたファイルのこと
- ❖ cmakeを使用することでOSに適したmakefileを自動で作成可能
- ❖ cmakeはCMakeLists.txtの内容に基づいて処理を行う
cmake [CMakeLists.txtが存在するディレクトリ]
- ❖ 実行する際はビルド用のディレクトリを用意するとプログラムの再配布などが容易になる

プログラム生成

❖ **make** コマンド：makefileの内容に従いプログラムを作成する

make [ターゲット名]

❖ 例：ターゲット「a.out」を作成するとき

make a.out

❖ 指定するターゲットはmakefileに記載されているものに限る

❖ ターゲット名を指定しない場合makefileの
一番初めに記載されているターゲットを実行する

プログラム実行

- ❖ 作成したファイルを実行するには,
./[実行ファイル名] コマンドを使用する
- ❖ 例：ディレクトリ /build/src/内の実行ファイルExecを実行する
 - ❖ カレントディレクトリがsrcの場合
./Exec
 - ❖ 絶対パスで実行する事も可能
/build/src/Exec

圧縮ファイルを解凍

❖ **unzip** コマンド：Linuxでzipファイルを解凍（展開）する

現在のディレクトリにファイルを解凍

unzip [解凍するzipファイル名]

解凍するディレクトリを指定

unzip [解凍するzipファイル名] **-d** [ディレクトリパス]

パスワード付きzipを解凍

unzip -P [パスワード] [解凍するzipファイル名]

文字列， 数値表示

❖ echoコマンド： ターミナル上へ文字列を表示するコマンド

文字列「HelloWorld」を表示する

```
echo HelloWorld
```

変数aの値を表示する

```
a = 10
```

```
echo $a
```

標準出力をファイルへ書き込む

❖ コマンドの出力結果を標準出力とよび、リダイレクトを使用することで標準出力をテキストなどのファイルへ書き込める

❖ ファイルに上書きする「>」

❖ 既ファイルと同名の場合、その中身を全て消すので注意

❖ 既ファイルに追記する「>>」

文字列「Hello」を同ディレクトリのText.txtに上書きする

```
echo Hello > ./Text.txt
```

OpenRTM-aistの概要

講義資料提供

❖ 国立研究開発法人産業技術総合研究所（産総研， AIST）

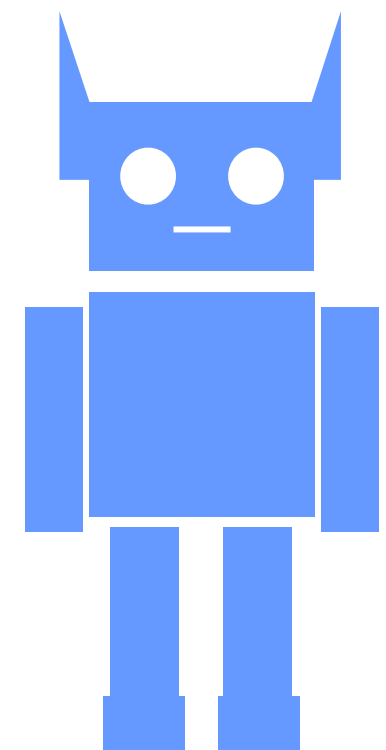
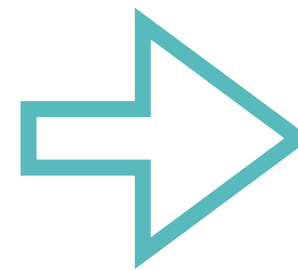
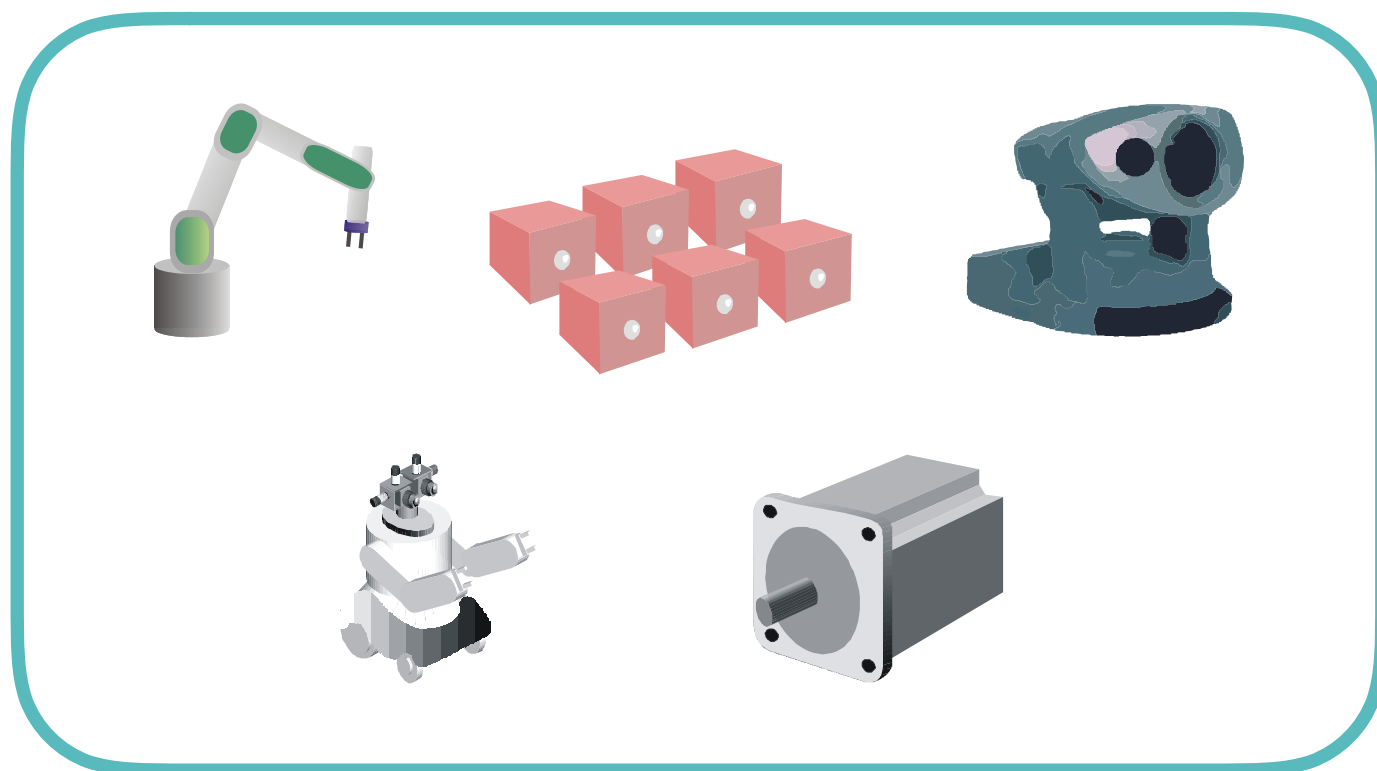
❖ ロボットイノベーション研究センター

<https://unit.aist.go.jp/rirc/>



RTとは?

- ❖ RT = Robot Technology (≠ Real Time)
- ❖ IT = Information Technology
- ❖ ロボット技術：センサ，アクチュエータ，制御などの集合体
 - ❖ 機械，電気・電子，情報を知る必要がある



RTMとは?

- ❖ RT-Middleware (RTM)

 - ❖ RT要素のインテグレーション(統合)のためのミドルウェア

- ❖ RT-Component (RTC)

 - ❖ RT-Middlewareにおけるソフトウェアの基本単位

産総研版 RTミドルウェア

OpenRTM-aist

従来のシステムでは



ジョイスティック



ロボットアーム



互換性がある
インターフェース同士は
接続可能

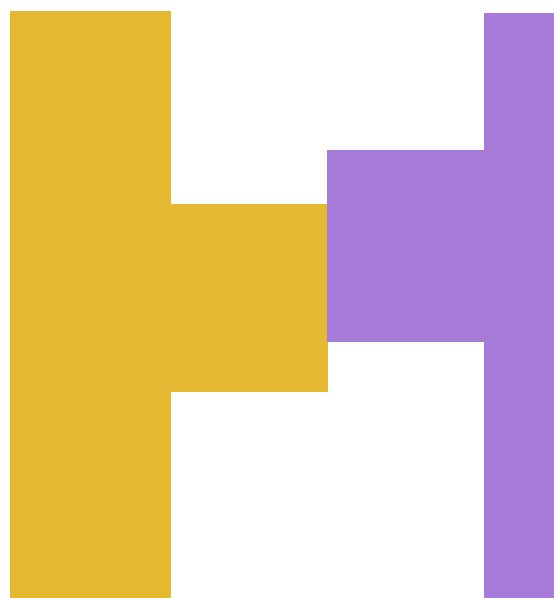
従来のシステムでは



ジョイスティック



ロボットアーム



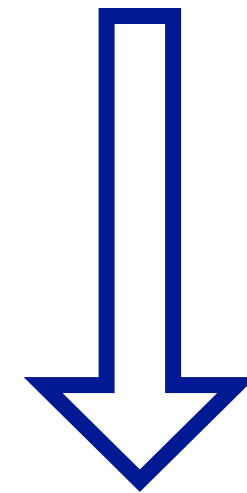
互換性がない
インターフェースは
つなげることはできない

RTミドルウェアでは



ジョイスティック

別々に作られたソフトウェア
モジュール同士をつなぐための
共通インターフェースを提供

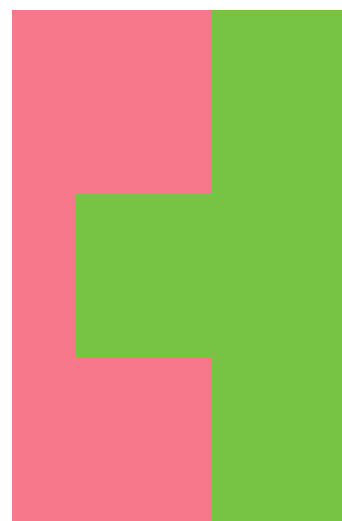


ソフトウェアの再利用性の向上により
RTシステム構築がより容易になる

RTミドルウェアでは



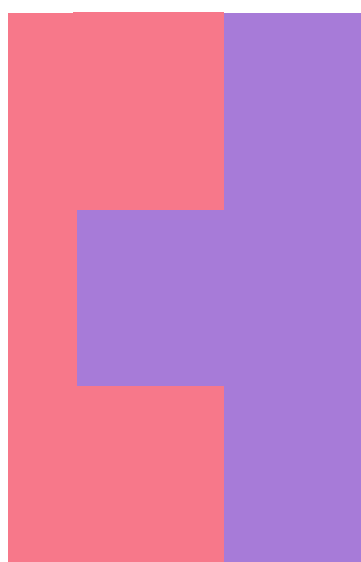
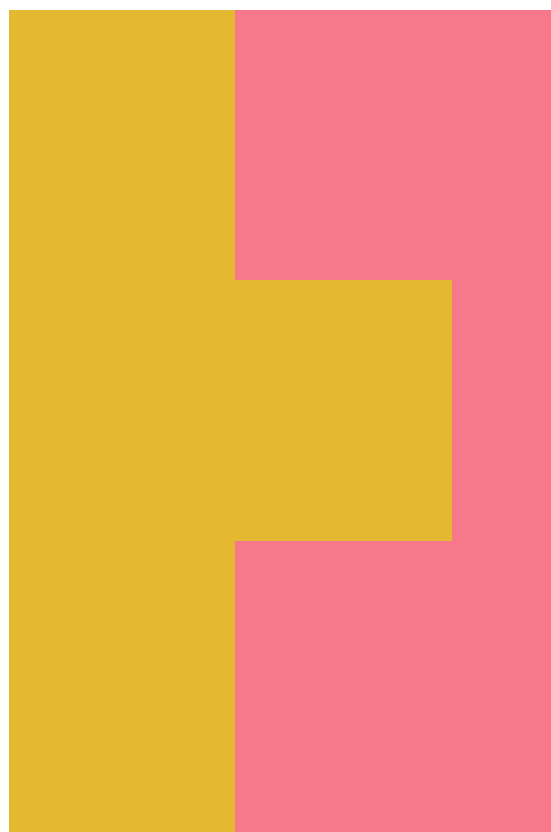
ジョイスティック



接続可能



ロボットアーム



接続可能

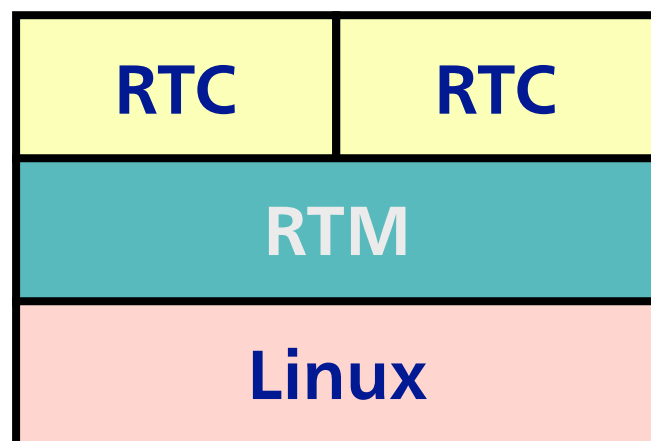


ヒューマノイド用
ロボットアーム

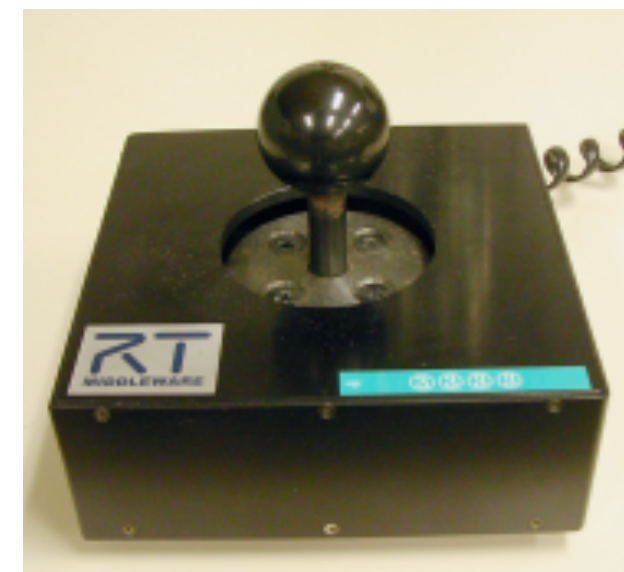
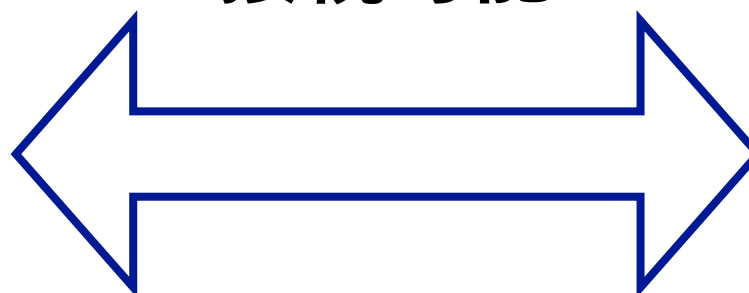
RTMによる分散システム



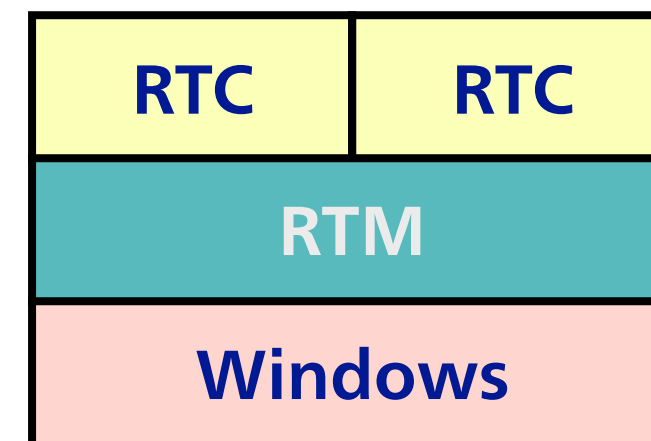
ロボット



接続可能



ジョイスティック

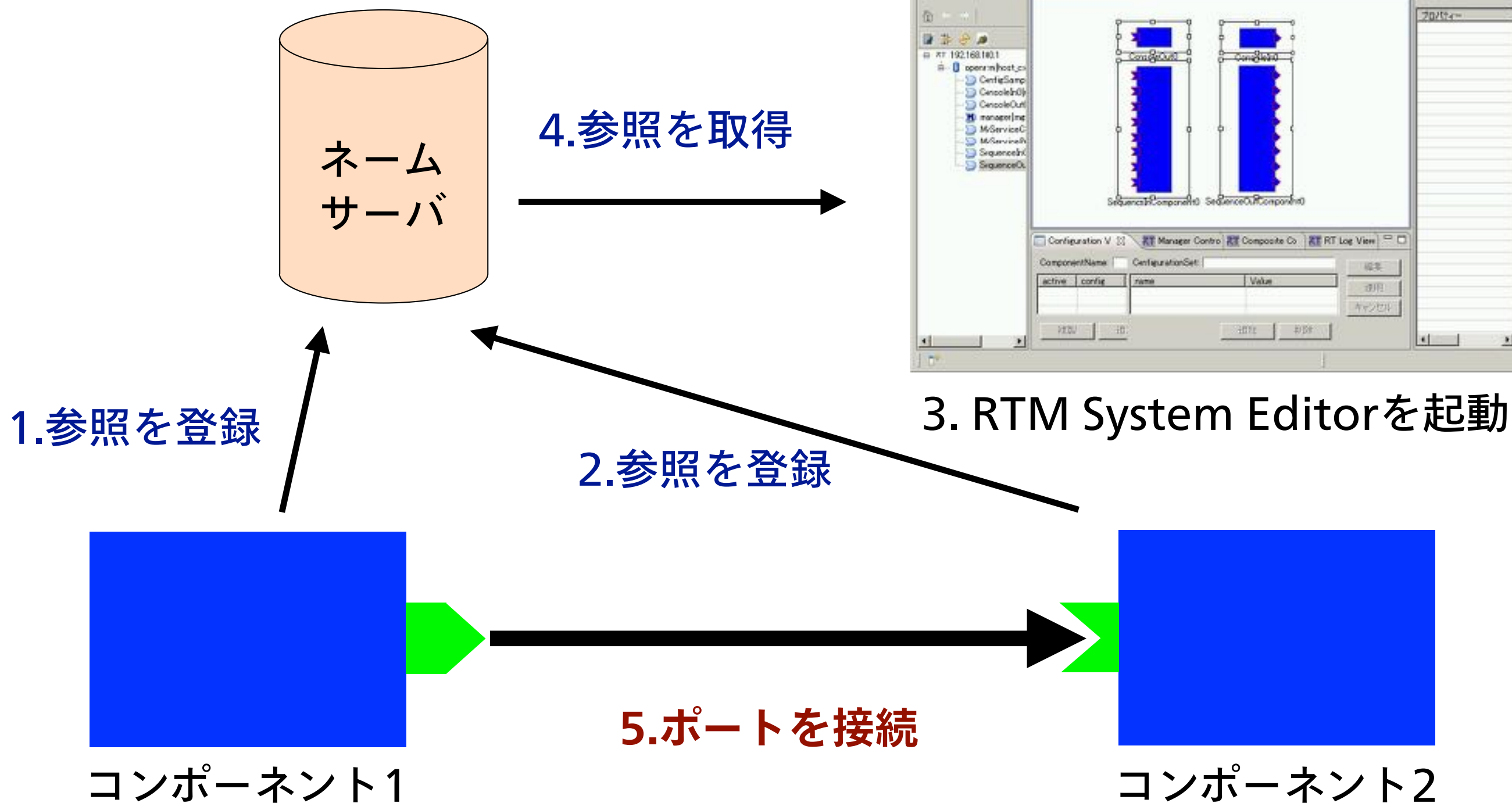


RTC同士の接続は
プログラム実行中に
動的に行える

RTMを使うことで、ネットワーク上に分散するRTCを
OS・プログラム言語の壁を越えて接続できる

動作シーケンス

0.ネームサーバを立ち上げる



OpenRTM-aistの 開発環境構築

インストールソフトウェアリスト

1. Python2.7.10
2. OpenRTM-aist 1.1.2-RELEASE
3. OpenCV
4. Pygame (※32bit版をインストール)

以下のサイトを参考にインストール

<https://tmp.openrtm.org/openrtm/ja/node/6026>

Python2.7.10インストール

1. Python2.7を以下URLからダウンロード
`https://www.python.org/ftp/python/2.7.10/python2.7.10.msi`
2. ダウンロードしたインストーラをクリックしてインストール
3. インストール時は, 「Add python.exe to Path」の設定を変更し, 「Will be installed on local hard drive」を選択

OpenRTM-aistインストール

1. インストーラを以下のURLからダウンロード

`http://tmp.openrtm.org/pub/Windows/OpenRTM-aist/1.1/
OpenRTM-aist-1.1.2-RELEASE_x86.msi`

2. ダウンロードしたインストーラをクリックして、インストール

OpenCVインストール準備

❖ numpyのインストール

1. [システム環境変数]の中から[Path]を探し以下があるか確認

➡ 無ければ, **;C:¥Python27;C:¥Python27¥Scripts** を追記

✓ C:¥Python27 : Python2.7インストールフォルダ

✓ C:¥Python27¥Scripts : Scriptsへのパス

※[システム環境変数]は,

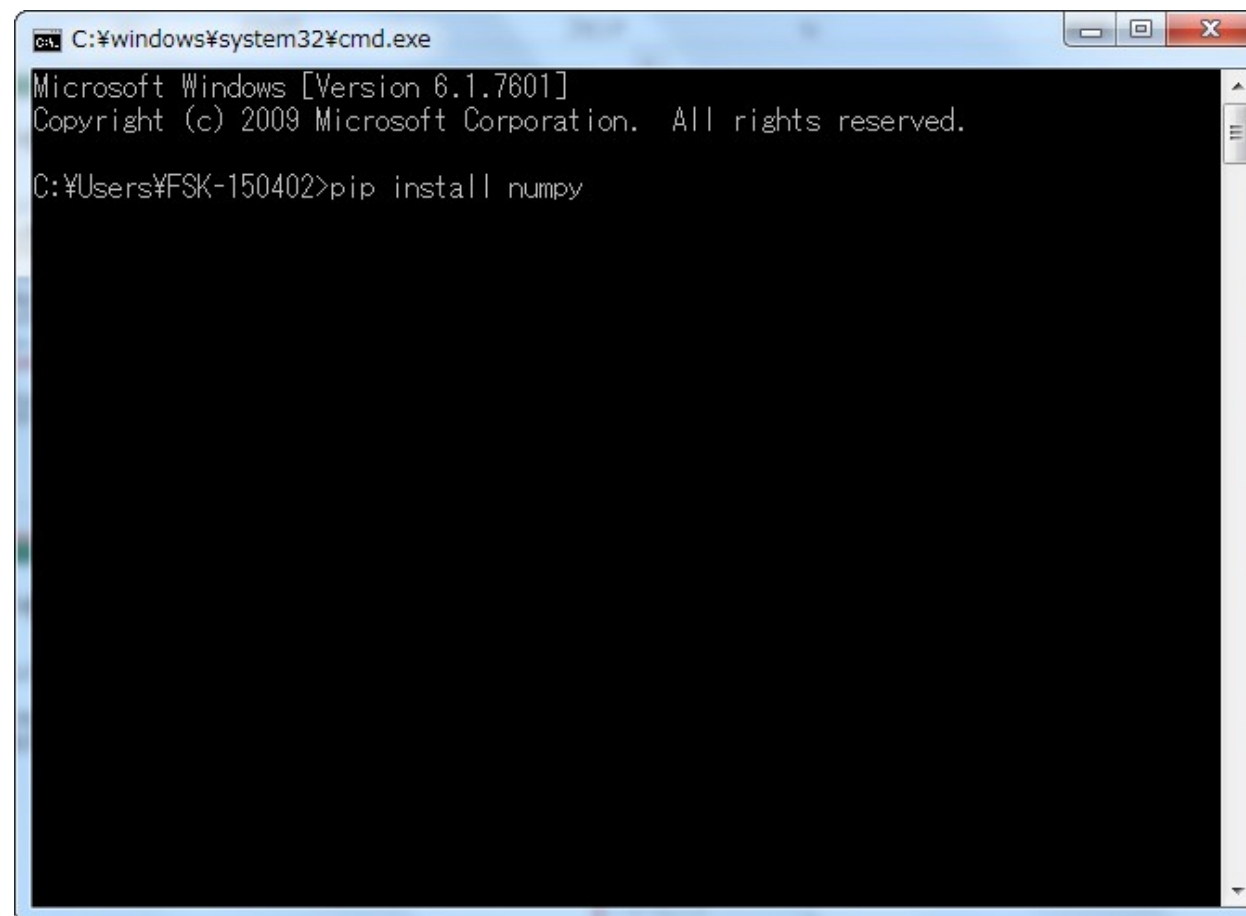
「コントロールパネル」→「システム」→

「システムの詳細設定」→「環境変数」でアクセスできる

OpenCVインストール準備

❖ numpyのインストール

2. コマンドプロンプト上で **pip install numpy** と打ち込む



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\FSK-150402>pip install numpy
```

「すべてのプログラム」 → 「アクセサリ」 →
「コマンドプロンプト」 で起動できる

OpenCVインストール

1. 下記のURLからOpenCVをダウンロードする
 - ➡ 「OpenCV library」 <http://opencv.org/>
 - ➡ 今回は2.4.11を使用
2. 上記のホームページにアクセスし、 ページ上部の [RELEASES] ボタンをクリック
3. ページ中段にある OpenCV 2.4.11 の [Win pack] をクリック
4. クリック後、 自動的にOpenCVのダウンロードが開始

OpenCVインストール

5. ダウンロードしたファイルを実行
6. フォルダを生成する場所を聞かれるので、好きな場所を指定
 1. この講習会では、C:¥ にフォルダを生成する
 2. C:¥ を指定すると、そこにOpenCVフォルダが作成
7. [Extract]ボタンをクリックして解凍が開始
8. 数分後、指定したフォルダ[opencv]が生成され、
[opencv]→[build]→[python]→[2.7]→[x86]と進み、
ファイル[cv2.pyd]をコピー (※64bitの場合[x64] 内に存在)
9. [Python2.7]→[Lib]→[site-packages]と移動し、
[cv2.pyd]を貼り付け

Pygameインストール

1. 以下URLへアクセス

<http://pygame.org/download.shtml>

2. Pygameをダウンロード

pygame-1.9.1.win32-py2.7.msiをダウンロード

3. Pygameをインストール ダウンロードした インストーラを クリックしてインストール

1.9.1 Packages (August 6th 2009)

Source

- [pygame-1.9.1release.tar.gz](#) ~ 1.4M - *source/docs/examples in unix format*
- [pygame-1.9.1release.zip](#) ~ 1.5M - *source/docs/examples in windows format*

Windows

Get the version of pygame for your version of python. You may need to uninstall old versions of pygame first.

NOTE: if you had pygame 1.7.1 installed already, please uninstall it first. Either using the uninstall feature - or remove the files: c:\python25\lib\site-packages\pygame . We changed the type of installer, and there will be issues if you don't uninstall pygame 1.7.1 first (and all old versions).

- [pygame-1.9.1.win32-py2.7.msi](#) 3.1MB
- [pygame-1.9.1release.win32-py2.4.exe](#) 3MB
- [pygame-1.9.1release.win32-py2.5.exe](#) 3MB
- [pygame-1.9.1.win32-py2.5.msi](#) 3MB
- [pygame-1.9.1.win32-py2.6.msi](#) 3MB
- [pygame-1.9.2a0.win32-py2.7.msi](#) 6.4MB
- [pygame-1.9.1.win32-py3.1.msi](#) 3MB
- [pygame-1.9.2a0.win32-py3.2.msi](#) 6.4MB
- (optional) Numeric for windows python2.5 (note: Numeric is old, best to use [numpy](#)) <http://rene.f0o.com/~rene/stuff/Numeric-24.2.win32-py2.5.exe>
- windows 64bit users note: use the 32bit python with this 32bit pygame.

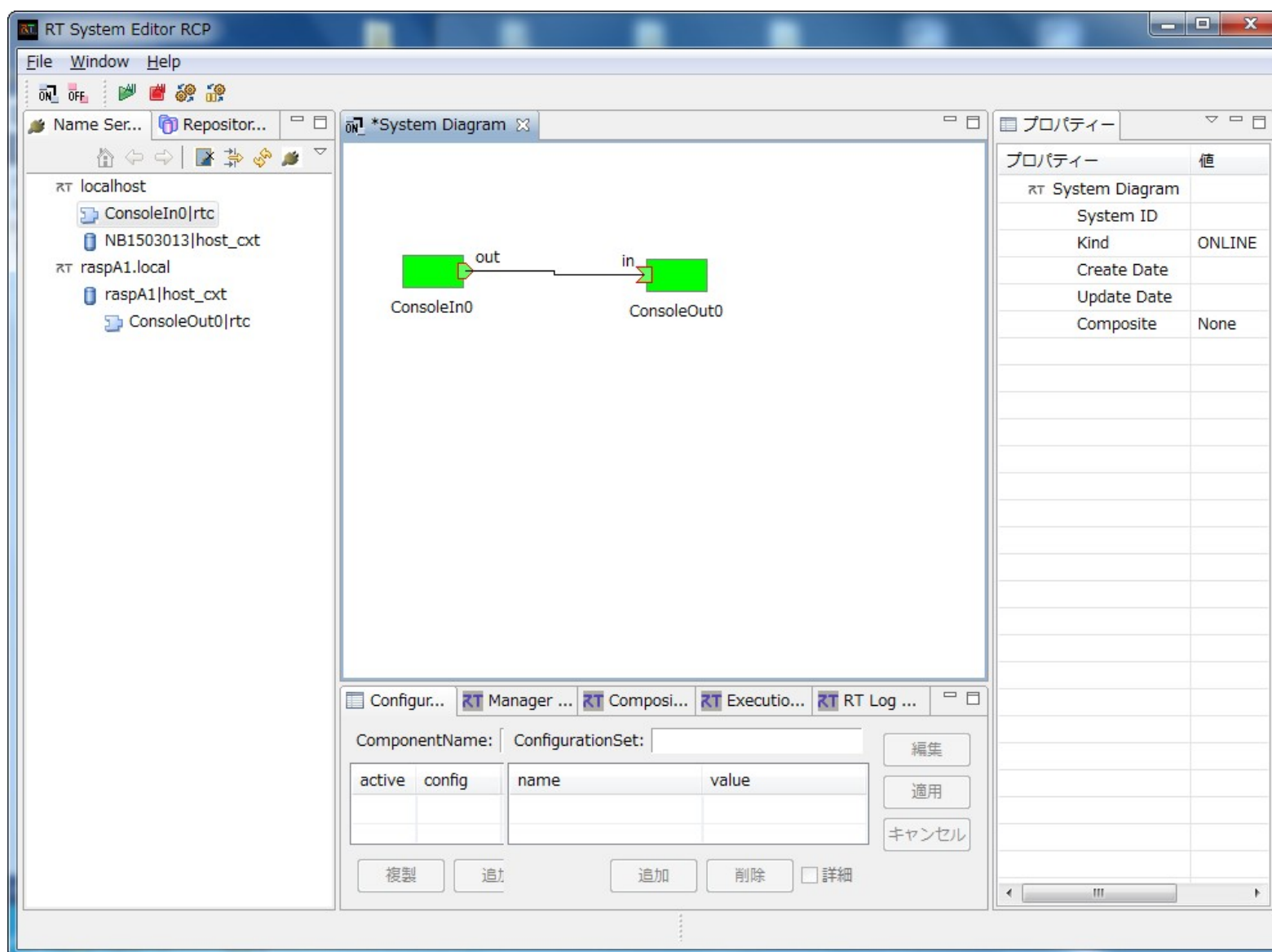
今日の課題

本日の課題

- ❖ OpenRTMを利用した遠隔操作ロボットを作成できる
 - 1. RTSystemEditorの使い方
 - 2. サンプルコンポーネントの実行
 - 3. Raspberry Piにカメラ接続
 - 4. ソフトウェアジョイスティックによる遠隔操作ロボット
 - 5. PS4コントローラによる遠隔操作ロボット

1. RTSystemEditorの使い方

❖ OpenRTMのツールの使い方, プログラムの起動の仕方を取得



2. サンプルコンポーネントの実行

- ❖ PCからRaspberry Piに数字を送信し，その数字を表示



数字入力

数字を送信



数字表示

3. Raspberry Piにカメラ接続

❖ Raspberry Piに接続したカメラから取得した画像をPCで表示



画像取得



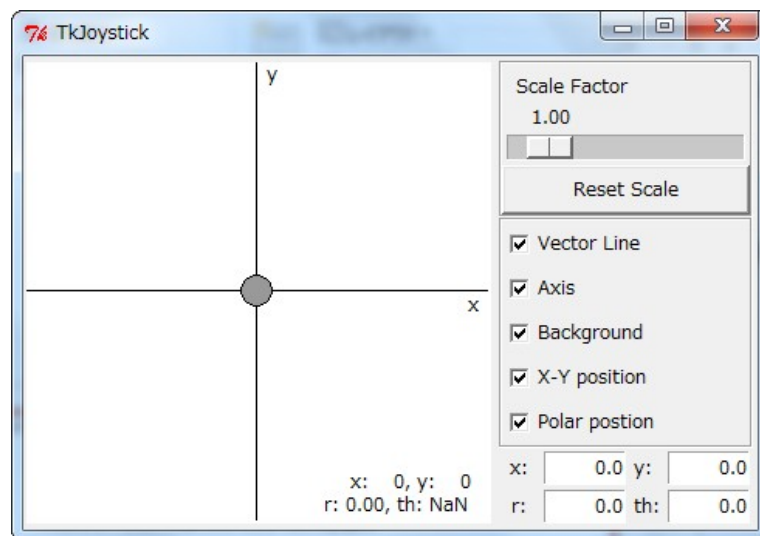
画像を送信



画像表示

4. ソフトウェアによるEV3操作

❖ PCのソフトウェアジョイスティックでEV3を操作



ソフトウェア
ジョイスティック

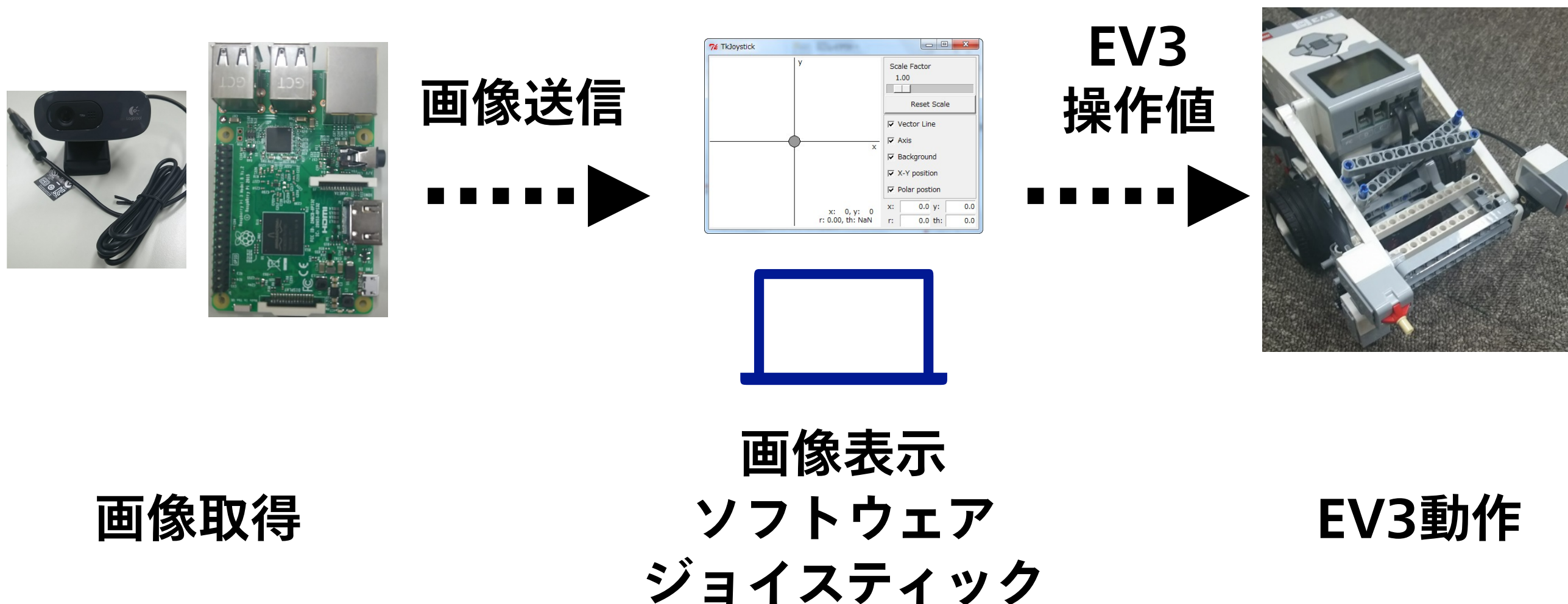
EV3操作の値を
送信



画像表示

5. ソフトウェアによる遠隔操作

- ❖ EV3上にRaspberry Piとカメラを乗せ、取得した画像を見ながらソフトウェアジョイスティックでEV3を操作



6. DualShock4による遠隔操作

- ❖ EV3上にRaspberry Piとカメラを乗せ、取得した画像を見ながらDualShock4(PS4コントローラ)でEV3を操作

