プログラミング応用

http://bit.ly/ouyou3d

オペレーティングシステムとプログラミング（1）

前期 第3週

2018/4/24

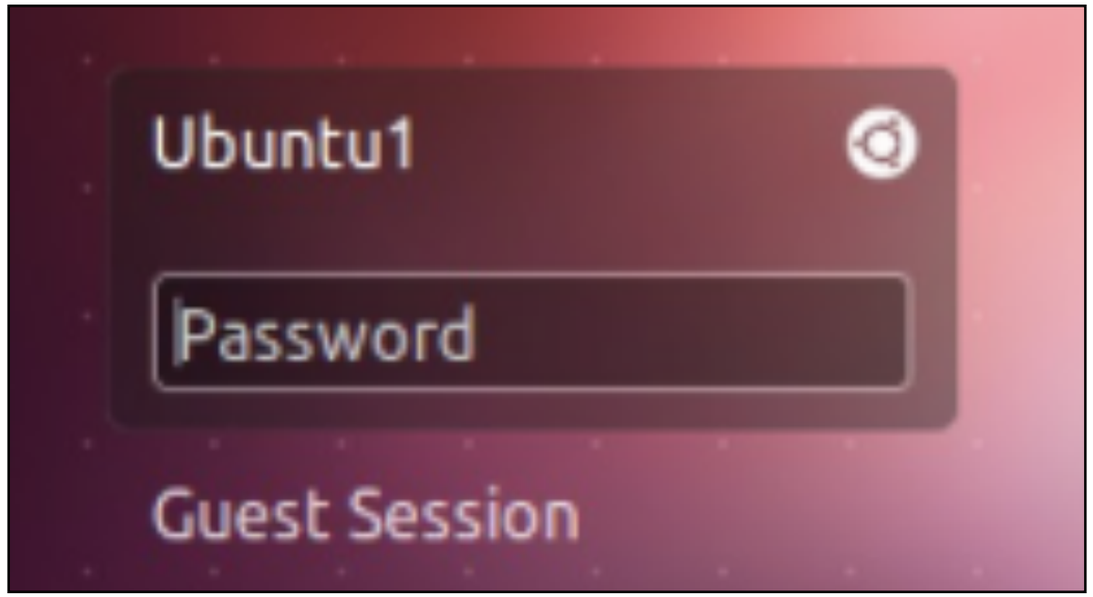
本日は・・・

- OSの役割
- OSの歴史
- OSが動作する仕組み
  - カーネル
  - プロセス, メモリ, ファイル
- 演習

  UNIXコマンドによるプロセス管理

  UNIXコマンドによるファイル管理

（例）ログイン用ソフトウェア



ログイン用ソフトウェアの基本動作は・・・

1. **ログイン用IDを入力**
2. **パスワードを入力**
3. **正しければデスクトップにジャンプさせる**

どうやって作る？

（例）ログイン用ソフトウェア

- ログイン用ソフトウェアの機能
  - キーボードから文字を受け取る機能
  - ディスプレイに受け取った文字を表示する機能

↓

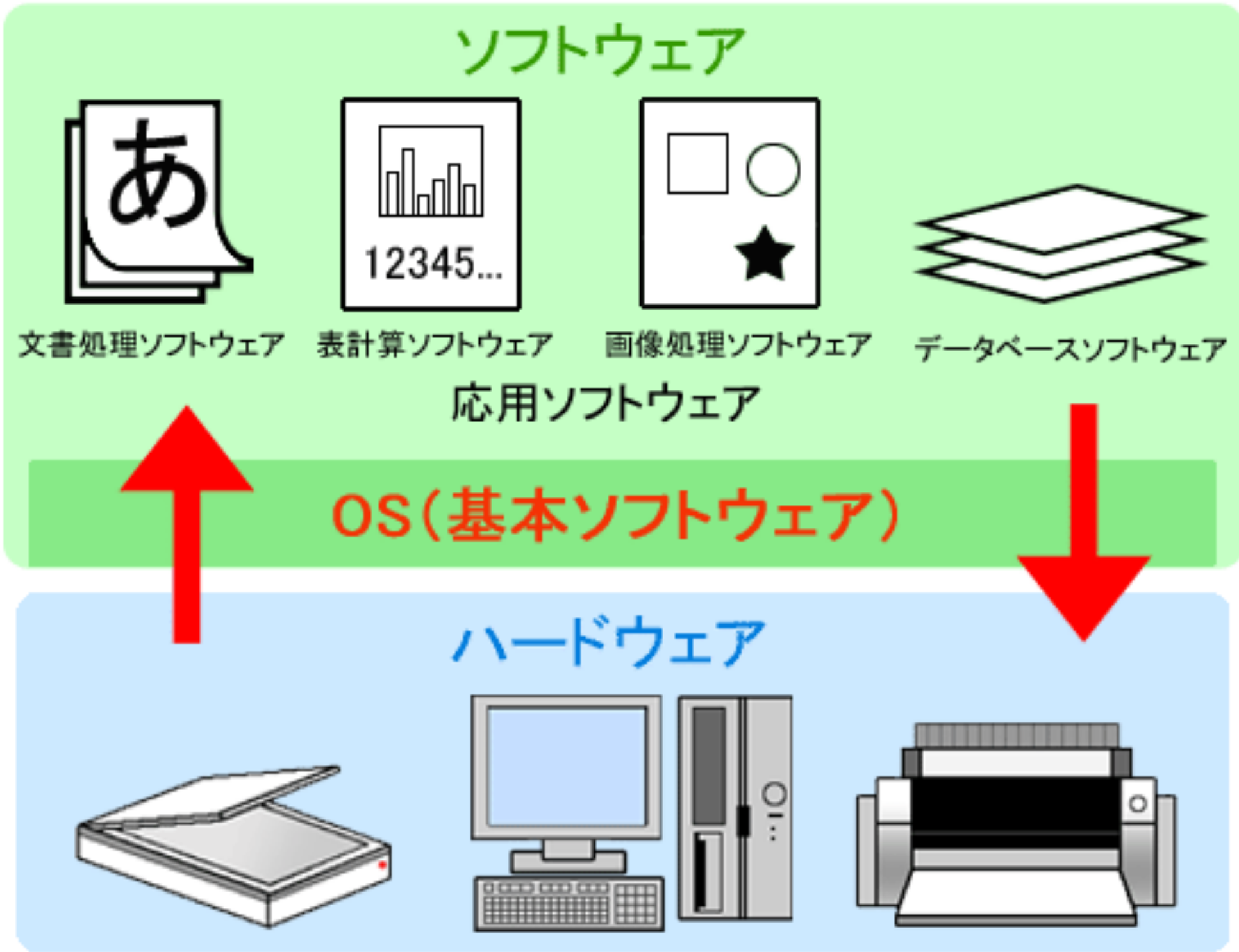
ハードウェアを制御する命令を記述する必要ない

- キーボード入力信号を検出
- ディスプレイのどのドットに何色を表示するか　など

↓

OS（オペレーティングシステム）が
ハードウェアを制御してくれる

ハードウェアとソフトウェア



OS代表的な役割

ハードウェアとソフトウェアの橋渡し役

- 計算資源の管理（＝プロセス管理）

  限られたCPUの計算能力を効率良く使えるように、

  プログラム実行の優先順位を付ける
- メモリ資源の管理（＝空間管理）

  限られたメモリ資源を有効に使えるように、

  プログラム実行に割り当てるメモリを調整
- ファイルシステムの提供（空間管理機能の一部）

  0と1で記録されたメモリ上の情報は人間には解読が困難のため、人間に分かりやすいファイル/ディレクトリといった概念で表現

OSの歴史（OS誕生以前）

- 1640–50年代：
  アプリケーションソフトウェアにハードウェアを制御する命令を記述
  - 機械語を0と1で記述
  - 開発者ソフトウェアもハードウェア制御も詳細に把握する必要があった
- 1960年中頃：
  IBM System/360にOS機能が確立し、複数プログラムが同時実行可能に

OSの歴史（OSの登場後）

- 1970年代：
  複数利用者が同時利用
- 1980年以降：
  GUI（グラフィカルユーザインタフェース）誕生
  - 個人用コンピュータとしてWindows(1985-) / MacOS(1984-)がシェアを独占
  - 開発者は、LinuxなどUNIXから派生したOSを利用することが多い
- 2000年代以降：
  モバイル端末向け軽量化されたUNIX系OSが誕生
  iOS/Android

OSの内部構造

- OS ＝ カーネルの集合体
  - 各々のカーネルはOSの基本機能を提供
  - カーネル提供する機能：プロセス管理、空間管理、ファイル管理、 割り込み制御、入出力制御、時間管理など
- アプリケーションソフトウェアは、システムコールを使ってカーネルの機能を呼び出す
  （アプリケーションソフトウェアは基本的にハードウェアを制御する命令を出すことは出来ない

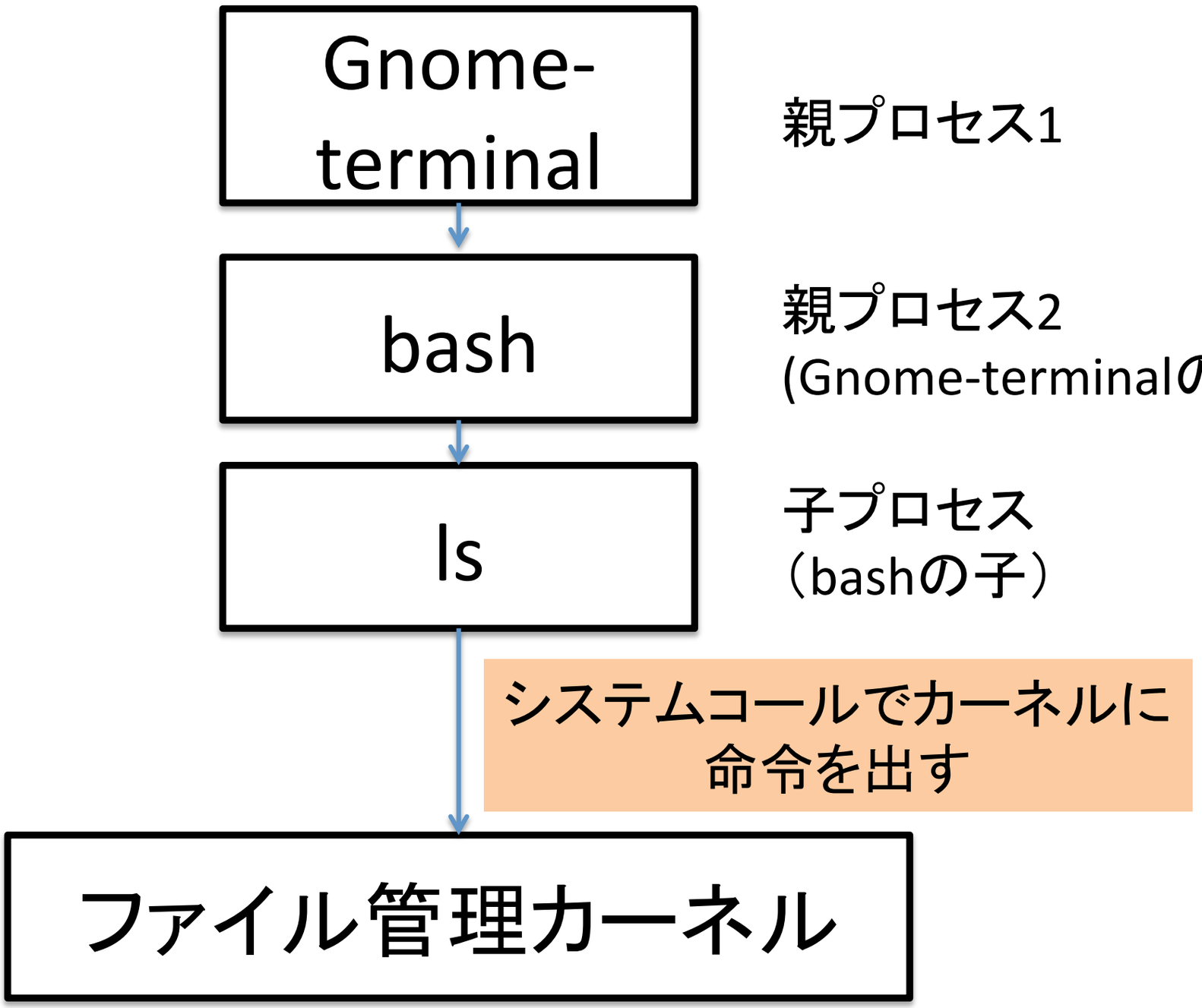
プロセス

プロセス ＝ プログラムの実行単位

［例］「端末」で ls コマンドを入力した場合

1. 「端末」を起動すると、gnome-terminalプロセスが起動
2. gnome-terminalは、bashプロセスを起動
3. lsコマンドを入力すると、lsプロセスが起動
4. lsプロセスは、ファイル管理カーネルに命令を出す
5. ファイル管理カーネルが画面に表示
6. lsプロセスが終了

子プロセスと親プロセス



プロセス管理

複数プロセスを同時に実行する仕組み

- CPUの計算資源は有限
  （同時実行可能なプロセス数＝CPUのスレッド数）
- OSは複数プログラムが効率的に実行されるよう実行するプロセスの優先順位を決定
- ユーザがアプリケーションを起動すると、1つ以上のプロセスが生成される

topコマンドやpsコマンドで実行中のプロセスを確認できる

空間管理

メモリ資源を管理する仕組み

- メモリ資源は有限
- どのプログラムにどの程度のメモリを割り当てるか決定する
- 実際には「仮想メモリ（Virtual Memory）」を用意して割り当てる
- プログラムがハードウェアメモリを使い果たすとハードディスクなどにデータを退避させる（＝スワップメモリ）

topコマンドやfreeコマンドで
メモリの使用状況を確認できる

ファイル管理

ファイル/ディレクトリの概念を提供

- メモリ上に0と1で情報が記憶
- ユーザはメモリのどこにファイルがあるか知らない
- ファイルシステム･･･メモリに記録されている情報を人間にも分かりやすい概念で表現

mkdir, ls, cd, cp, mvなどコマンドで
OSの提供するファイル操作を行うことが出来る

ファイル操作のための主なコマンド

- フォルダを作る
  `$　mkdir　フォルダ名`
- フォルダを削除する
  `$　rmdir　フォルダ名`
- フォルダを移動する
  `$　cd　フォルダ名`
- 1つ上のフォルダに移動する
  `$　cd　..`（ピリオド2個）
- ホームフォルダに移動する
  `$　cd　~/`

- フォルダに含まれるファイル（フォルダ）の一覧を表示する
  `$　ls`　（エル・エス）
- 中身が空のファイルを作る
  `$　touch　ファイル名`
- ファイルを削除する
  `$　rm　ファイル名`
- ファイルをコピーする
  `$　cp　コピー元ファイル名　コピー先ファイル名`
- ファイルを移動する（名前を変更する）
  `$　mv　移動元ファイル名　移動先ファイル名`

OSの機能の呼出し方

- **シェル(端末)から呼び出す**

  本日の演習（UNIXコマンドを用いたプロセス管理）

- **アプリケーション・プログラムから呼び出す**

  →C言語などからシステムコールを使って呼ぶ

  次回以降の内容（複数UNIXコマンドを組み合わせて使う、C言語からカーネルに命令を出す、など）

【課題3-1】

端末を開いて、topコマンドを実行してみましょう。
この情報から、以下のことを調べて試して下さい。

- topコマンドで表示されるプロセスをCPU使用率順に並べ替えて表示するには？
- topコマンドで表示されるプロセスをメモリ使用順に並べ替えて表示するには？
- 「bash」というプロセスの、プロセスID、実行しているユーザ、CPU使用率、メモリ使用は？

【課題3-2】

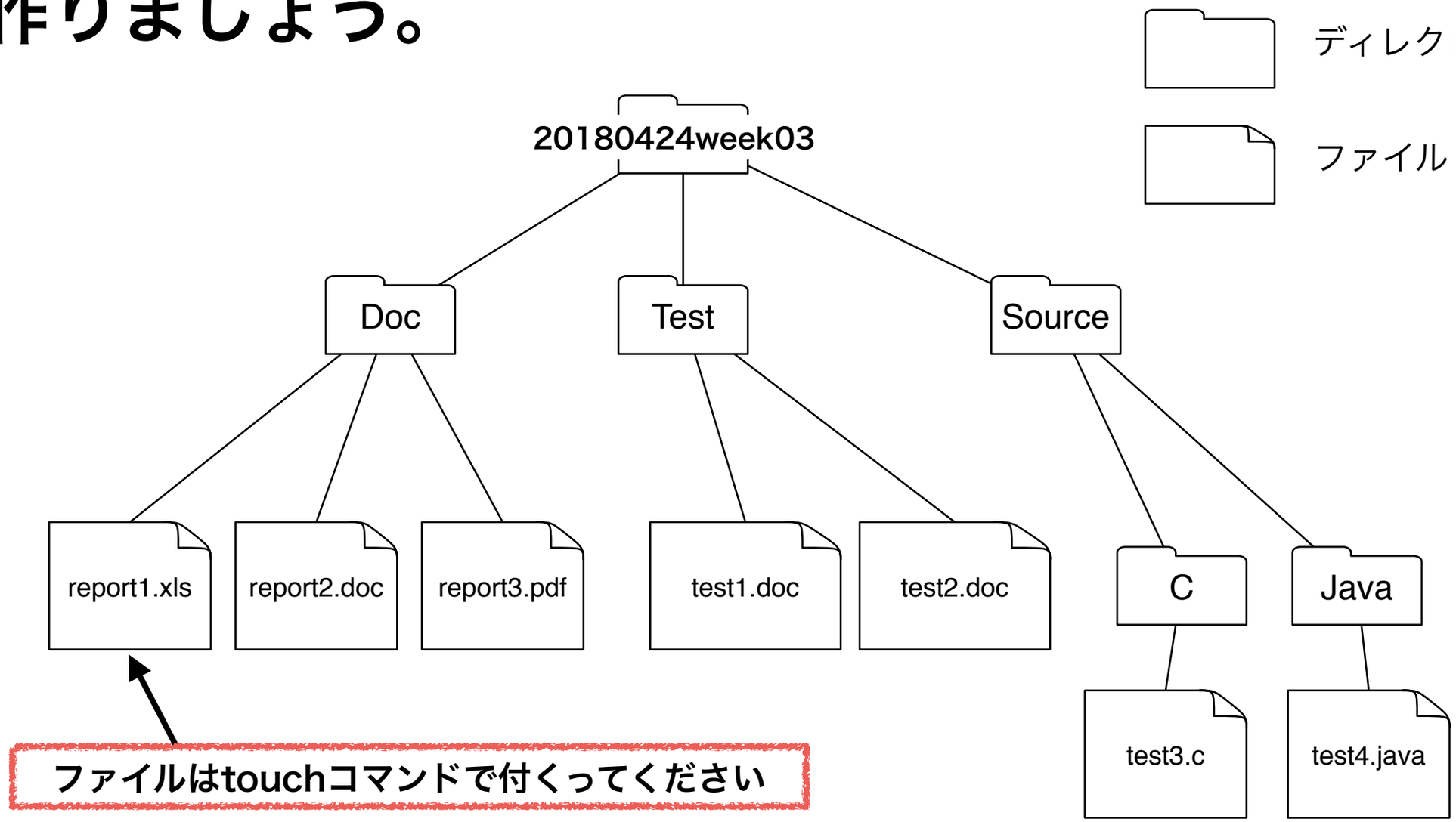
端末を開いて、ptreeコマンドを実行してみましょう。
この情報から、bashプロセスの親子関係がどのよう
になっているのか確認して下さい。

【課題3-3】

プロセスを終了するコマンドkillの使い方を調べて、bashプロセスを終了させてみて下さい。

【課題3-4】

ファイル操作のコマンドを使って、自分のホームディレクトリ以下に、次のようなファイルの階層構造を作りましょう。



【課題3-4】（つづき）

作ったファイルの階層構造は、次のようなlsコマンドで、フォルダに含まれる全てのファイルを表示できますので、同じ結果になるか確認してみましょう

```
$ ls -R ~/20180424week03
/home/st○○d○○/20180424week03:
Doc  Source  Test

/home/st○○d○○/20180424week03/Doc:
report1.xls  report2.doc  report3.pdf

/home/st○○d○○/20180424week03/Source:
C  Java

/home/st○○d○○/20180424week03/Source/C:
test3.c

/home/st○○d○○/20180424week03/Source/Java:
test4.java

/home/st○○d○○/20180424week03/Test:
test1.doc  test2.doc
```