並列プログラミング
Parallel Programming

2022 2Q

演習　第6回

情報理工学院　情報工学系

本日の流れ

- 課題内容の説明
- 演習に取り組む

演習課題概要

- **目的**
  - OpenCL の並列計算を体験
- **題材**
  - 数値列の計算
  - 画像加工フィルタプログラム

課題のダウンロード

以下からダウンロードしてください

学内アクセス

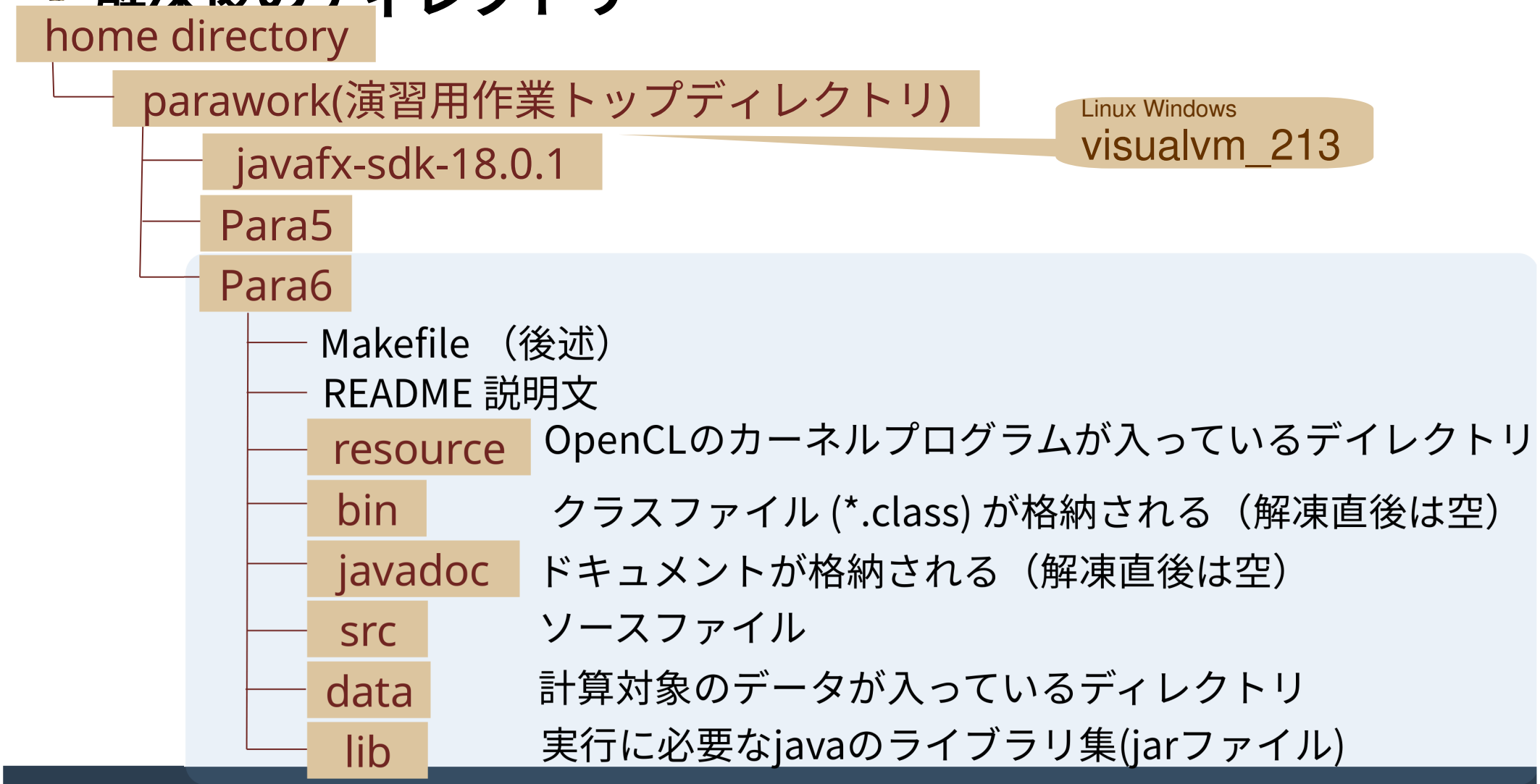
- www.img.cs.titech.ac.jp/lecture/para/

準備（1)

- ダウンロードした para6.zip を展開する

```
unzip para6.zip
```

- 解凍後のディレクトリ



準備（2)

- Para6 以下の ._* ファイルを再帰的に消去

```
cd Para6
make cleanall
```

MacOS では、 ._ で始まるファイルが作成されることがありますが、コンパイル時の障害になるので消去

準備（3)

- **ソースファイルを javac コマンドでコンパイルしてクラスファイルを作る**

今回の演習ではトップディレクトリで

```
javac -d bin -encoding UTF-8 --module-path ../javafx-sdk-18.0.1/lib
--add-modules javafx.controls,javafx.swing -sourcepath src -classpath
lib/*: src/para/Main12.java
```

Main12.javaが他のクラスに依存する場合，順次コンパイルしてくれる

Windowsでは : を ; に置換

実際は一行で書く

として下さい

`-d bin`	コンパイル後のクラスファイルをディレクトリ bin に置く
`-encoding UTF-8`	ソースファイルの文字コードが UTF-8 であることを示す
`-sourcepath src`	ソースファイルがディレクトリ `src` 以下にあることをコンパイラに教える
`--module-path`	モジュールファイルがあるディレクトリをコンパイラに教える
`--add-modules javafx.control`	依存するモジュールを列挙する
`-classpath`	依存するクラスやライブラリのありかをコンパイラに教える

※ https://docs.oracle.com/javase/jp/17/docs/specs/man/javac.html
でその他のオプションを確認すること

準備（4)

- **java コマンドでクラスファイルを実行する**
  - 今回の演習ではトップディレクトリにて

```
java --module-path /opt/Java/JavaFX/javafx-sdk-18.0.1/lib --add-modules
javafx.controls,javafx.swing -cp bin:lib/*: para.Main12
```

パッケージ名 起点となるクラスの名前

Windowsでは : を ; に置換

として下さい（デモ用プログラムは para.Main?? と para.opencl.HelloWorld para.opencl.Max3 para.opencl.Sum3 があります）

`--module-path` javacのオプションと同じ役割

`--add-modules` javacのオプションと同じ役割

`-cp bin:lib/*:`

-cp は -classpathの短縮形
実行に必要なコンパイル済みクラスファイルがディレクトリbin 以下に置かれていること、標準以外のjavaライブラリファイル(jarファイル)がlib/に置かれていることを java コマンドに教える
※ https://docs.oracle.com/javase/jp/17/docs/specs/man/java.html
でその他のオプションを確認すること

準備 (5)

- javadoc コマンドでソースファイルのコメント文から HTML のドキュメントファイルをつくる

```
package para.paint;

import javafx.application.Application;
        ⋮
/** JavaFXで作成するお絵描きプログラム.*/
public class Paint extends Application
{
    /** 描画領域.*/
    Canvas canvas;
```

HTML文書の出力先ディレクトリ

今回の演習では、 Para6 直下

実際は一行で書く

```
javadoc -html5 -charset utf-8 -encoding UTF-8 --frames -d javadoc
-sourcepath src
--module-path /opt/Java/JavaFX/javafx-sdk-18.0.1/lib --add-modules
javafx.controls,javafx.swing
-link https://docs.oracle.com/javase/jp/17/docs/api
-link https://openjfx.io/javadoc/18
-package para.opencl para para.graphic.shape para.graphic.target
para.graphic.parser para.graphic.camera para.graphic.opencl
```

外部javadocのURIリンクを作成するには必要

パッケージ名

としてください

※ https://docs.oracle.com/javase/jp/17/docs/specs/man/javadoc.html でその他のオプションを確認すること

準備 (5)

- **コマンドをいちいちタイプするのが面倒 ...**

今回は Makefile を用意したので make コマンドで
javac , javadoc の実行が簡単に行える

`make Main12`	Main12 をコンパイルして、実行
`make Main13`	Main13 をコンパイルして、実行
`make Main14`	Main14 をコンパイルして、実行
`make HelloOpenCL`	HelloOpenCL をコンパイルして、実行
`make Sum3`	Sum3をコンパイルして、実行
`make Max3`	Max3 をコンパイルして、実行
`make clean`	bin 以下のクラスファイルをすべて削除
`make doc`	javadoc コマンドを実行

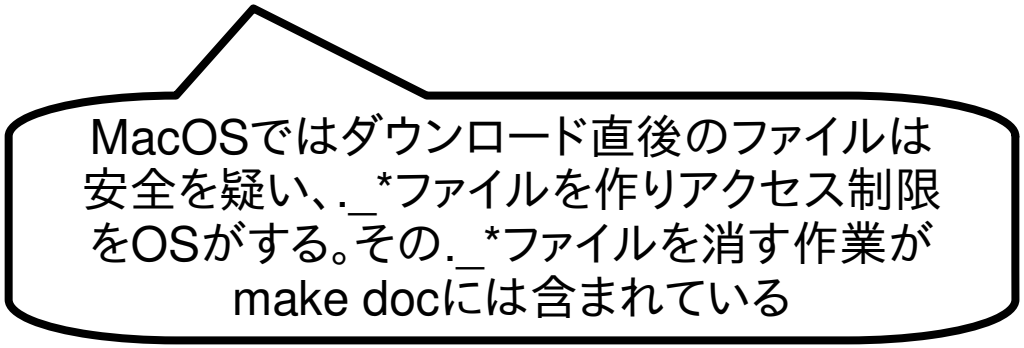
ソースコードを更新してもmakeが感知しない場合があるので、コードを書き換えても結果に変化がない場合は、一度 make clean して再コンパイルして下さい

上を実行すると実際に発行されたコマンドが表示される
Makefile を自分好みに変更してよいです

Makefileの記述ではタブ\tは意味があります。スペースで置き換えると、makeが正しく解釈できません。
Makefileの書き方は各自調べて下さい

課題 0

- javadoc コマンドを実行して HTML 文書を生成し，ブラウザで閲覧する
  - Mac OS X では `open HTML ファイル名` とすればブラウザが起動する
  - 各クラスのパッケージ名などを確認する
  - ブラウザのエンコーディングの設定は UTF-8 にする
  - コンパイルエラーが起こる場合は展開直後にトップディレクトリで一度 make doc とタイプすると解決する場合がある



課題 1

OpenCLを用いて、data/ にあるdataa.txt datab.txt datac.txtの数値に対して演算を行い、結果を標準出力に出力せよ

```
%java -cp bin:lib/*:resource
para.opencl.Max3
8.0 8.0 12.0
%
```

Max3

```
%java -cp bin:lib/*:resource
para.opencl.Sum3
10.0 3.0 12.0
%
```

Sum3

1.1)dataa.txt datab.txt datac.txtのそれぞれn番目の要素をa[n]、b[n]、c[n]としたとき、

- a[n]+b[n]+c[n]を計算して順に標準出力へ出力するプログラム para.opencl.Sum3
- a[n],b[n],c[n]の中の最大値を順に標準出力へ出力するプログラム para.opencl.Max3

をそれぞれ完成させよ。ただし、para.opencl.HelloOpenCLを参考にして計算はOpenCLを使うこと。ただしカーネルプログラム中ではif文のような条件分岐を使うと並列計算が遅くなる。、そこで条件分岐を極力用いずにOpenCL C言語の組み込み関数を利用した実装を考えること。

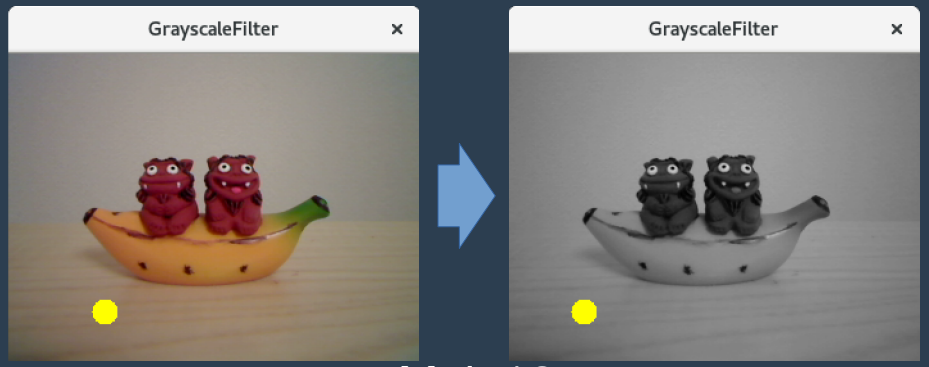
Sum3.java Max3.javaの雛形は src/para/opencl にある。

hint OpenCL C言語の組み込み関数については仕様書もしくはクイックレファレンスカードを参照すること

プログラムの完全なソースコードと実行に必要なファイルを提出ディレクトリにコピーすること（OpenCLのカーネルプログラムのコピーを忘れないこと）

課題 2

para.Main12はカメラ映像をOpenCLを用いて各画素の輝度を計算し、モノクロ映像にして表示するプログラムの未完成品である。
プログラムを完成させ、モノクロ映像を表示させよ。



Main12

2.1)Main12で用いているgray.cl のGrayカーネル関数は各画素のカラーの値から輝度を計算して無彩色に変換する処理するための雛形である。gray.clを書き換え、プログラムを完成させよ。
輝度の計算には

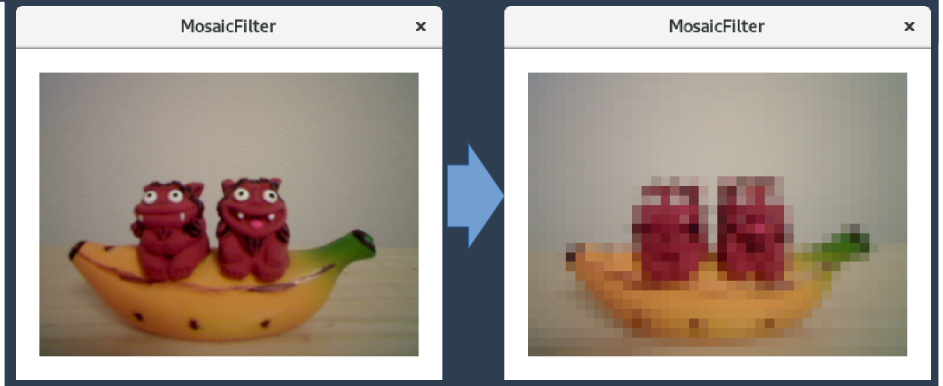
輝度=赤の強度*0.212671 + 緑の強度*0.715160 + 青の強度*0.072169

の式を用い、その値を無彩色の値とせよ。

hint 無彩色にするには赤、緑、青に同じ値を入れればよい。
hint 映像データから輝度を正しく計算するにはガンマ補正についての変換、逆変換が必要であるが、今回の課題では省略して良い。
hint カーネル関数にカメラから来るデータは1画素あたり3バイトであるが、処理後は1画素あたり4バイトとしている。増えた1バイトは透明度情報である。
hint 表示される動く黄色い丸は課題には関係がない。
hint ホストプログラムの変更は不要である。

課題 3

配布したpara.Main13はカメラ映像にモザイク処理をするプログラムの未完成版である。「8x8の画素を1ブロックとし、各ブロックをその中の画素の平均色で塗りつぶす」という処理によりモザイク処理をOpenCLを用いて施すようにして完成させよ。



Main13

3.1) resource/para/mosaic.cl がカーネルプログラムのファイルである。
1ワークグループが1ブロックの描画に対応している。1ブロックに対応する画素をローカルメモリにコピーする部分と、ローカルメモリの先頭番地からの3つで表わされる色の値をグローバルメモリに書き出す部分は用意されている。平均色を計算する部分を埋めて与えられた仕様のモザイク処理を行うプログラムを完成させよ。

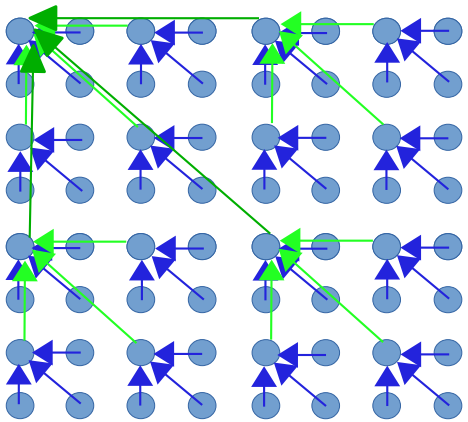
hint 講義で説明した和の計算法を応用せよ

hint ワークグループ内での作業の同期を取るため

```
barrier(CLK_LOCAL_MEM_FENCE);
```

を必要箇所に挿入せよ。

hint ホストプログラムは変更不要である。

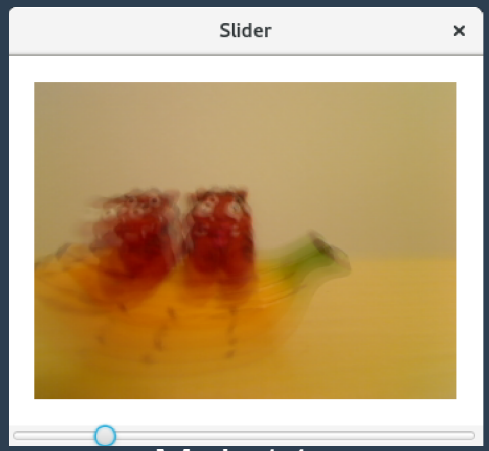


8x8の画素値の集約

3.2) このプログラムでの1ワークグループに所属するワークアイテムの個数を答えよ。

課題 4

配布したpara.Main14は、映像に残像効果を与え、映像を表示しつつファイルに録画するプログラムの未完成版である。プログラムを完成させよ。なお与えられた式を用いてスライダにより残像の強さを調整可能とすること。



Main14

4.1) スライダにより設定される値を0≦d≦1とする。1つ前の時刻で表示した画素の値をp、現在時刻のカメラの画素の値をnとする時、現在時刻の画素に表示される値を h = n*d+p*(1-d) とし、p=h としてpを毎時刻更新する。この演算法をresource/para/delay.clを作成して実装せよ。

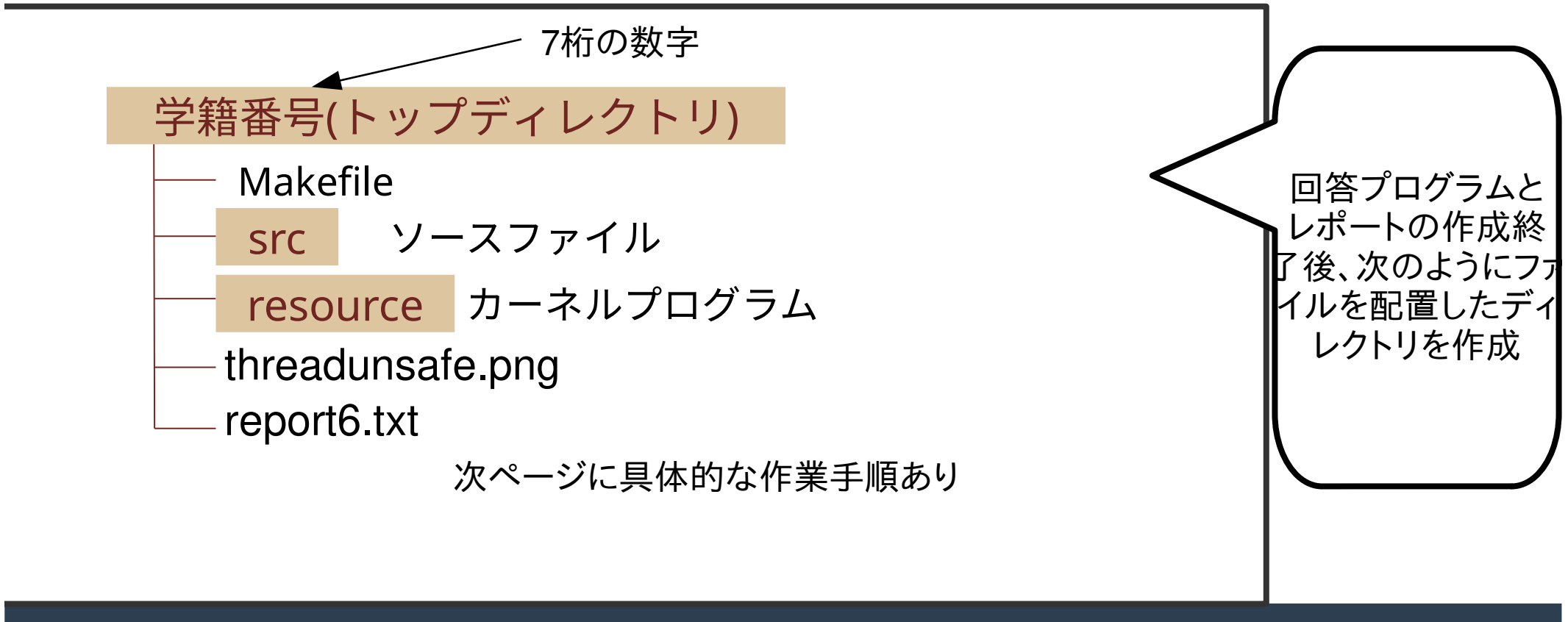
hint カメラ画像は1画素3バイト形式、処理の出力は1画素4バイト形式なので、演算しやすいように1つ前の時刻の出力の1画素3バイト形式のバッファも用意する

4.2) para.Main14を実行するとrecorddelay.mp4という映像ファイルが作成される。この録画映像には壊れた静止画像が記録されることがある。実はプログラムのどこかがスレッドセーフとしては不十分な設計のため生じている。スレッドセーフでないことで生じたことが明瞭に分かる映像をpara.graphic.shapeパッケージの図形をアニメーションさせることで作成せよ。映像中、最も端的な瞬間を静止画像としてキャプチャし、threadunsafe.pngとして回答に含めよ。

4.3) スレッドセーフでない原因について説明せよ。さらに、その原因を取り除いてスレッドセーフな録画機能を実現せよ。実現のための設計の改善方針について説明をし、ソースコードの修正部分を示せ。

提出方法（1 of 3)

- **para6.zip を展開したディレクトリ構造を保ったまま，課題 の変更作業を行う**
- **各課題で自分が変更したファイルの先頭には自分の名前と学籍番号を書いておく**
  - プログラムの場合はコメント内に書く
- **課題 1 から 4 の回答文、工夫点および感想を書いた report6.txt を用意する (雛形は課題のウェブページ)**



提出方法（2 of 3)

- **提出用ディレクトリを作成する**

  学籍番号から7桁の数字にすること

```
mkdir   dir
```

- **ソースファイルのディレクトリのコピーを作る**

  今回はPara6

```
cp -R トップディレクトリ/src トップディレクトリ/resource   dir
```

- **dir に Makefile report6.txt もコピーする**

```
cp トップディレクトリ/Makefile トップディレクトリ/report6.txt dir
```

- **dir に課題４で作成した画像データをコピーする**

```
cp トップディレクトリ/threadunsafe.png dir
```

- **次のコマンドを実行する**

  学籍番号に対応する7桁の数字にすること

```
zip ex6-2012345.zip -r dir
```

- **dir 以下の内容が圧縮され、ex6-2012345.zip が作られます**

  圧縮後に内容を“ unzip ex6-2012345.zip” で確認すると提出ミスを防げて安全

提出方法 (3 of 3)

- **作成した zip ファイルを T2SCHOLA にアップロードする**
- **締め切り**

  7 月 28 日（木）　23:59 (JST)