**Computer Systems Technology**

British Columbia Institute of Technology

COMP 8005 - Assignment2- Design

Albert Huang

Feb 21, 2018

# Table of Contents

[Table of Contents 1](#_Toc1159748805)

[Design 2](#_Toc1799238149)

[1.1 Multi-Threading 2](#_Toc1140181709)

[1.1.1 Flowchart 2](#_Toc580218177)

[1.1.2 Description 3](#_Toc641307704)

[1.1.3 Pseudo Code 4](#_Toc1632175190)

[1.2 Multi-Processing 6](#_Toc1422342552)

[1.2.1 Flowchart 6](#_Toc632593838)

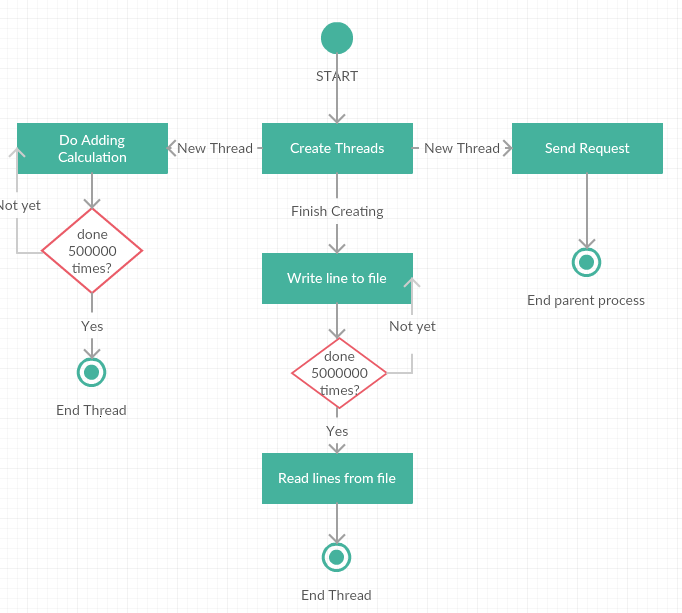
[1.2.2 Description 7](#_Toc1172114812)

[1.2.3 Pseudo Code 8](#_Toc1229738628)

# Design

## 1.1 Multi-Threading

### 1.1.1 Flowchart



### 

### 1.1.2 Description

1. For each test, it will generate a bunch of threads
2. The main stream would wait for every threads ended before the next test starts

Using thread join([time]) method.

1. Each thread would do the CPU-bound test

Each thread would do 1,500,000 times adding calculation before ending.

1. Each thread would do the IO-bound test

Each thread would write a line for million times to a file and then read it to a list.

1. Each thread would do the HTTP request test

Each thread would send request for a web page and receive the data.

1. After each test the app would print the time consuming

The main stream code will print time consuming which comes from time recording.

### 1.1.3 Pseudo Code

// create a list for working threads

// record the current time

// Loop and create thread

//--- child thread---

// While counter < 500,000

// counter + 1

// do x += x calculation

// do another x += x calculation

// End While

// End child thread

// print the time consuming

// create a new list for working threads

// record the current time

// Loop and create thread

//--- child thread---

// While counter < 5,000,000

// write a fixed line to file

// End While

// Read all lines from that file to a list

// End child thread

// print the time consuming

// create a new list for working threads

// record the current time

// Loop and create thread

//--- child thread---

// Send HTTP request to a website

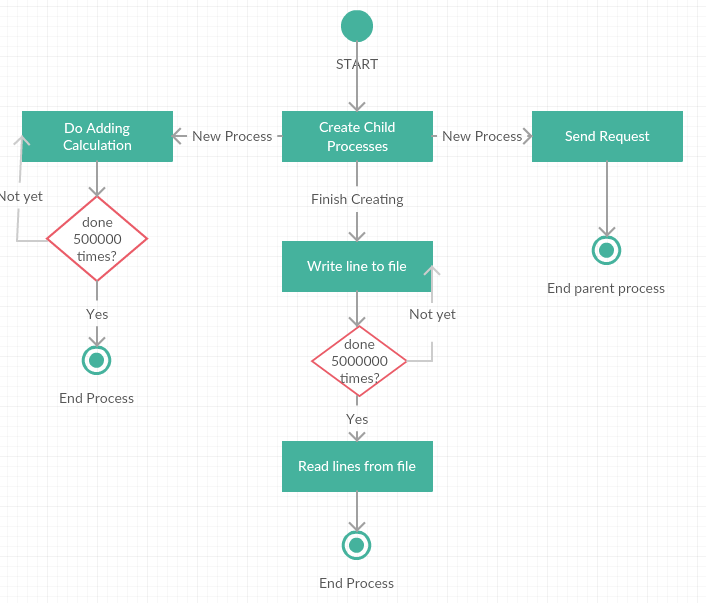
// receive the data or throw Exception

// End child thread

// print the time consuming

## 1.2 Multi-Processing

### 1.2.1 Flowchart



### 

### 1.2.2 Description

1. For each test, it will generate a bunch of child processes
2. The main stream would wait for every threads ended before the next test starts

Using thread join([time]) method.

1. Each process would do the CPU-bound test

Each process would do 1,500,000 times adding calculation before ending.

1. Each process would do the IO-bound test

Each process would write a line for million times to a file and then read it to a list.

1. Each process would do the HTTP request test

Each process would send request for a web page and receive the data.

1. After each test the app would print the time consuming

The main stream code will print time consuming which comes from time recording.

### 1.2.3 Pseudo Code

// create a list for child processes

// record the current time

// Loop and create processes

//--- child process---

// While counter < 500,000

// counter + 1

// do x += x calculation

// do another x += x calculation

// End While

// End child process

// print the time consuming

// create a new list for child processes

// record the current time

// Loop and create child processes

//--- child process---

// While counter < 5,000,000

// write a fixed line to file

// End While

// Read all lines from that file to a list

// End child process

// print the time consuming

// create a new list for child processes

// record the current time

// Loop and create child processes

//--- child process---

// Send HTTP request to a website

// receive the data or throw Exception

// End child process

// print the time consuming