

Programmazione Concorrente e Distribuita - Assignment 02

Andrea Biagini
Filippo Gurioli
Leonardo Randacio

**Università di Bologna
Scienze e Ingegneria Informatiche**

April 20, 2024

Contents

1	Macropart 1	1
2	Macropart 2	2
2.1	Task Decomposition	2
2.2	Event-loop	3
2.3	Reactive	3
2.4	Virtual Threads	3

List of Figures

Chapter 1

Macropart 1

Chapter 2

Macropart 2

2.1 Task Decomposition

The main task can be represented in pseudocode with a recursive function `getWordOccurrences(str URL, str WORD, int DEPTH)`

The function identifies two tasks:

- `countTask`
- `linkTask`

The `countTask` finds all occurrences of the defined word in the website, while the `linkTask` finds all the links and for every link starts the `getWordOccurrences` on the new url found. Both subtasks are expressed below:

Algorithm 2.1: countTask expressed in pseudocode

```
1 input: str SITE_CONTENTS, str WORD
2 output: int
3 begin
4   int COUNT ← 0
5   foreach CURRENT_WORD in SITE_CONTENTS
6     if (CURRENT_WORD == WORD)
7       COUNT ← COUNT + 1
8     endif
9   endforeach
10  return COUNT
11 end
```

Algorithm 2.2: linkTask expressed in pseudocode

```
1 input: str SITE_CONTENTS, str WORD, int DEPTH
2 begin
3   if (DEPTH != 0)
4     foreach CURRENT_WORD in SITE_CONTENTS
5       if (is_url(CURRENT_WORD))
6         getWordOccurrences(CURRENT_WORD, WORD, DEPTH - 1)
7       endif
8     endforeach
9   endif
10 end
```

2.2 Event-loop

Source code: TODO ADD THE SOURCE CODE DIR The tasks are represented by events. The first function call creates an event that parses through the contents of the given website. If a url is found and the depth value is not 0 a new event is created, that executes the function on the new website, with the depth value decreased by one.

2.3 Reactive

Source code: TODO ADD THE SOURCE CODE DIR TO BE DONE

2.4 Virtual Threads

Source code: TODO ADD THE SOURCE CODE DIR The function creates a virtual thread that executes the following pseudocode:

```
1 input: str URL, str WORD, int DEPTH
2 begin
3   str SITE_CONTENTS ← get_site_contents(URL)
4   int COUNT ← 0
5   if (DEPTH != 0)
```

```

6      foreach CURRENT_WORD in SITE_CONTENTS
7          if (CURRENT_WORD == WORD)
8              COUNT ← COUNT + 1
9          endif
10         else if (is_url(CURRENT_WORD))
11             VT = create_new_vt( () ->
12                 getWordOccurrences(CURRENT_WORD, WORD, DEPTH - 1))
13             join(VT)
14         endforeach
15     endif
16 end

```
