

Parallel Scheduling Plan

Team Members and Contribution:

Dalton Lim (dlim779) – All Stages

Chuyang Chen (cche381) – All Stages

Raghav Garg (rgar969) – All Stages

Magnus Rand (cran263) – All Stages

Brian Nguyen (bugn877) – All Stages

JULY 29 2019

Course: SOFTENG 306



Tea6

Table of Contents

Introduction3

Work Breakdown Structure Diagram4

 Work Breakdown Structure Diagram Figure5

Network Diagram6

 Network Diagram Figure7

Gantt Chart.....10

 Gantt Chart Figure11

Introduction

This document outlines the steps involved in the first project of the SOFTENG306 course. Our models are in accordance to the waterfall models. Here the following phases are outlined, starting with a planning phase, followed closely by coding, testing and then the management tasks.

To help maximize the extent to which we follow the waterfall model, the planning stage is exhaustive in describing the tasks required for each stage as well as their sequential order. This is done through three diagrams:

- The Work Breakdown Structure (WBS) Diagram:
A brainstorm of which tasks each of the four waterfall stages entail
- The Network Diagram:
A portrayal of functional dependencies found within our brainstormed tasks
- The Gantt chart:
A representation of the temporal dependencies and the resources each task might require.

Each diagram is also accompanied with text containing a description, justification and explanation of the diagram.

Work Breakdown Structure Diagram

The Work Breakdown Structure Diagram is a visualization of the tasks required to complete the project. Our diagram contains four sections involved within the Waterfall model: Planning, Coding, Testing and Management/Miscellaneous tasks. This diagram exists for the purpose of being a comprehensive list of all the piece of work required; which becomes a useful reference during the creation of the later network diagram and Gantt chart. Therefore, the existence of this diagram is crucial for the initial planning of the project as all future planning depends on this diagram.

Project: Using AI and parallel processing power to solve difficult scheduling problem

Planning

Workflow Brainstorm - Identify the tasks involved in the completion of the project

Time estimate for tasks - Estimating how long each piece of work will take

Requirements - Finding the specifications for the project deliverables

Categorization of tasks - Putting the tasks into groups as to better fit the waterfall model

WBS-diagram - Construction of the Work Breakdown Structure Diagram

Gantt Chart - Construct a Gantt chart using the time estimates and WBS network diagram

Network diagram - Order the items on WBS diagrams based on which items need to be complete before others

Group work split up appropriately - Split groupwork so that items of work can be done efficiently and quickly

A strategy for the visualisation - discuss and choose how a visual representation of the solution is possible

Heuristics For implementable algorithms - Discuss and analyse each workable algorithm and decide which ones are more viable, and efficient, than others

A plan using the Waterfall model - Develop a roadmap for using the waterfall model which completes the desired tasks on time

UML Diagram - Specify, visualize, construct and document artifacts of a software system

Coding

Estimate of execution time - Estimating how quickly and efficiently the implemented algorithm(s) run

Workable input processsing - Creation of a system which can take formatted input data and transform it into a corresponding graph representation

Multithreaded Code - The project should be runnable on more than one thread to decrease time taken to form a solution

Executable JAR file - A runnable JAR file containing the project solution

Implementation of most efficient algorithm - Creation of a system which uses the best algorithm to solve the project problem.

Improve efficiency - Decrease the time taken for a generated solution as much as possible.

Minimise error frequency & severity - Fix all possible instances of errors within the system.

Correct Output file - The system should produce the correct syntax

Testing

Test cases - construct and execute test cases which test whether or not the system runs in the correct state

Solution compatible with Linux - Solution must work on the Linux OS

Solution compatible with windows - Solution must work on the Windows OS

Management

Documentation and Wiki - Updating the Changelog, Notes, and Wiki on Github

Presentation - The presentations which must be done in lectures as an update on our progress

Report - Writing the report which comes at the end of the project.

Interview - Presenting the current state of the system to the lecturer

Network Diagram

The Network Diagram is a flow chart which provides an overview of functional dependencies between tasks, outlining the chronological order of tasks as well as groups of tasks which can be completed concurrently. The tasks in the diagram are extracted from the WBS diagram and their dependencies are analyzed to chronologically order them. The individual tasks are colour-coded according to the section it originated from. This diagram makes task scheduling easier and enables us to quickly gauge when parallelization of our tasks may be used. This will help to ensure we are completing the project as efficiently as possible.

Gantt Chart

The Gantt Chart is a representation of the amount of time and resources needed to complete each task. This diagram is a natural extension of the Network Diagram, as it contains not only the functional dependencies of tasks, but also the estimates of the resources and time which need to be dedicated for each task. The existence of this diagram is therefore crucial for a structured execution of the project in accordance to the waterfall model. This model will also play a major role in the time management of the project, helping outline exactly when each task should be finished.

GANTT CHART

[illegible]