**Mezemir Gebre**

**Parallel Programming skills foundation individual report**

**Race condition:**

(2p) What is race condition?

**It is a condition that happens when two different tasks that must be carried out sequentially or in a certain time difference happen to go at the same time interrupting the execution process in a system or device.**

(5p) Why race condition is difficult to reproduce and debug?

**It is unpredictable and unrepeatable and may not exhibit themselves during the debugging and reproduction time because for a failure to occur**, the **timing of events must be exactly right**.

(8p) How can it be fixed? Provide an example from your Project\_A3 (see spmd2.c)

**Since it is difficult to reproduce and debug, we should design our programs carefully rather than trying to fix it when it happens. In spmd2.c program, the cores share one bank of memory because of the variables declared outside of the block that will run in parallel so we fix it by declaring the variables inside the block that will be forked and run in parallel in separate threads so we can avoid sharing one bank of memory which can possible cause race condition in a system.**

- (15p) Summaries the Parallel Programming Patterns section in the “Introduction to Parallel Computing\_3.pdf” (two pages) in your own words (one paragraph, no more than 150 words).

**Parallel programs have many documented patterns created by other developers for a future new developer who may want to utilize that. These patterns grouped in to two as Strategies and Concurrent Execution Mechanisms. Algorithmic strategies and implementation strategies are two types of strategies. Algorithmic strategies are about deciding and figuring out which tasks should run concurrently by multiple processing units. While implementation strategies deal with how this algorithm and its components should be structured or implemented. Concurrent Execution Mechanism divided in to two as process/thread control and coordination patterns. The first patterns concerned with controlling the processing units of parallel execution on the hardware at run time. While the second one set up different simultaneously running tasks on processing units to complete parallel Conditional patterns have two major coordination Patterns, which are message passing between concurrent processes and mutual execution between threads executing concurrently on one bank memory system.**

(12p) In the section “Categorizing Patterns” in the “Introduction to Parallel Computing\_3.pdf” compare the following:

Collective synchronization (barrier) with Collective communication (reduction):

**They both are categories of coordination, which is a category under concurrent execution mechanism.**

Master-worker with fork join: **These two are different program structures categorized under the implementation strategies of parallel patterns to structure or implement our parallel algorithm that we came up with.**

(26p) **Dependency**: Using your own words and explanation, answer the following:

(3p) Where can we find parallelism in programming?

**We can find parallelism on the tasks that assigned to processors and on the way the data computed and stored by the processors.**

(6p) What is dependency and what are its types (provide one example for each)?

**Dependency is the interconnectedness of different operations to produce an output.**

(3p) When a statement is dependent and when it is independent (Provide two examples)?

**1.When two statements can be executed exchangeable in different order, they are independent.**

**2. if the next statement takes some inputs from the previous statement to produce an output then it is a dependent statement.**

(3p) When can two statements be executed in parallel?

**If they are independent of each other.**

(3p) How can dependency be removed?

**Dependency can be removed by getting rid of some statements or by repositioning statements.**

(8p) How do we compute dependency for the following two loops and what type/s of dependency? **By identifying any relationship between the variables used in the statements and by analyzing any variable that could be fetched or modified from the memory. This is a flow dependency because the first statement on both loops is dependent on the variable i also a is changing as i changes. The second statement on the right loop is Independent because it is not dependent on the first statement. It can be executed in different order without any effect.**