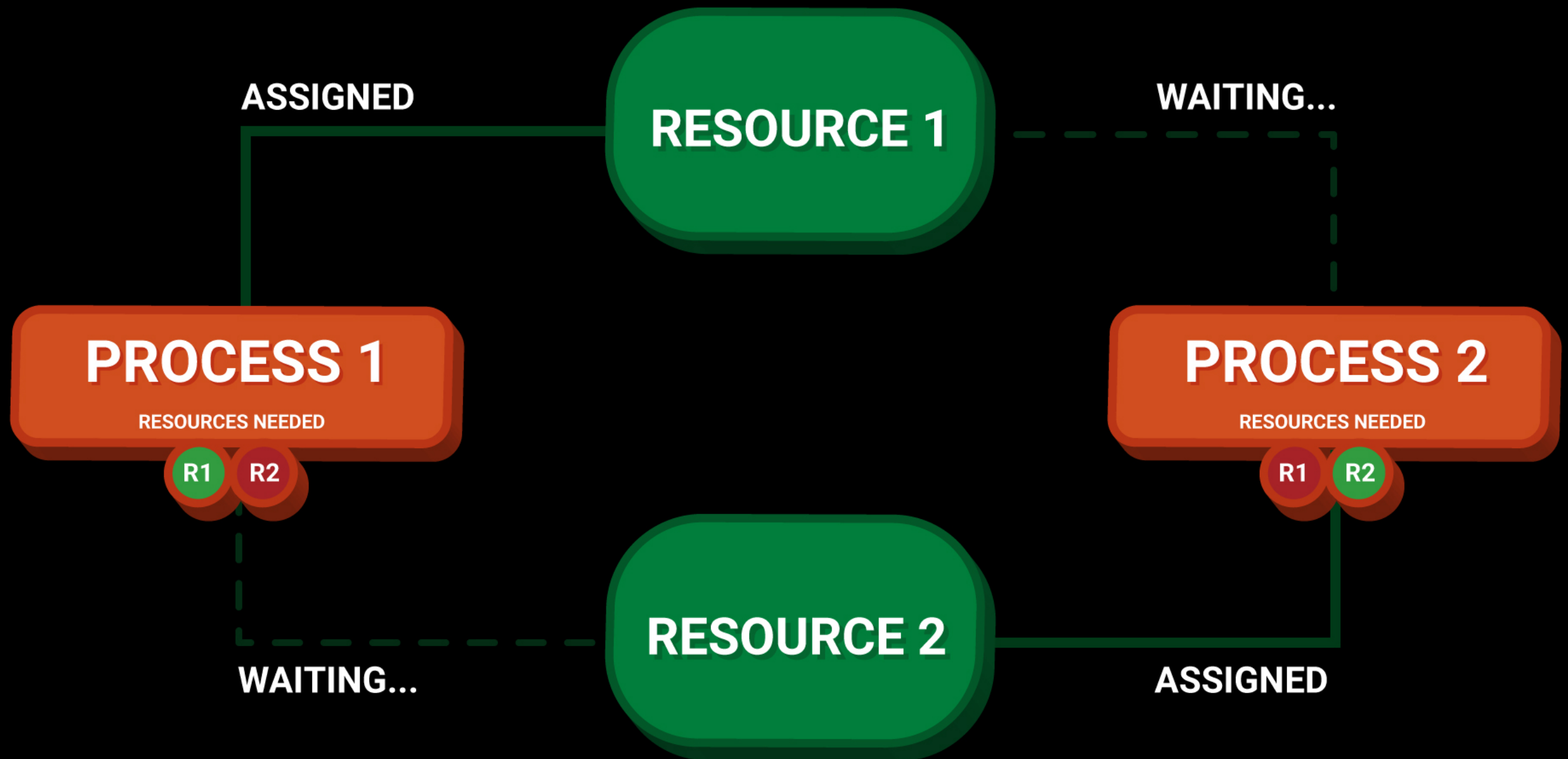M Bedir Tapkan

DINING PHILOSOPHERS

Operating System Presentation
North American University – 2017
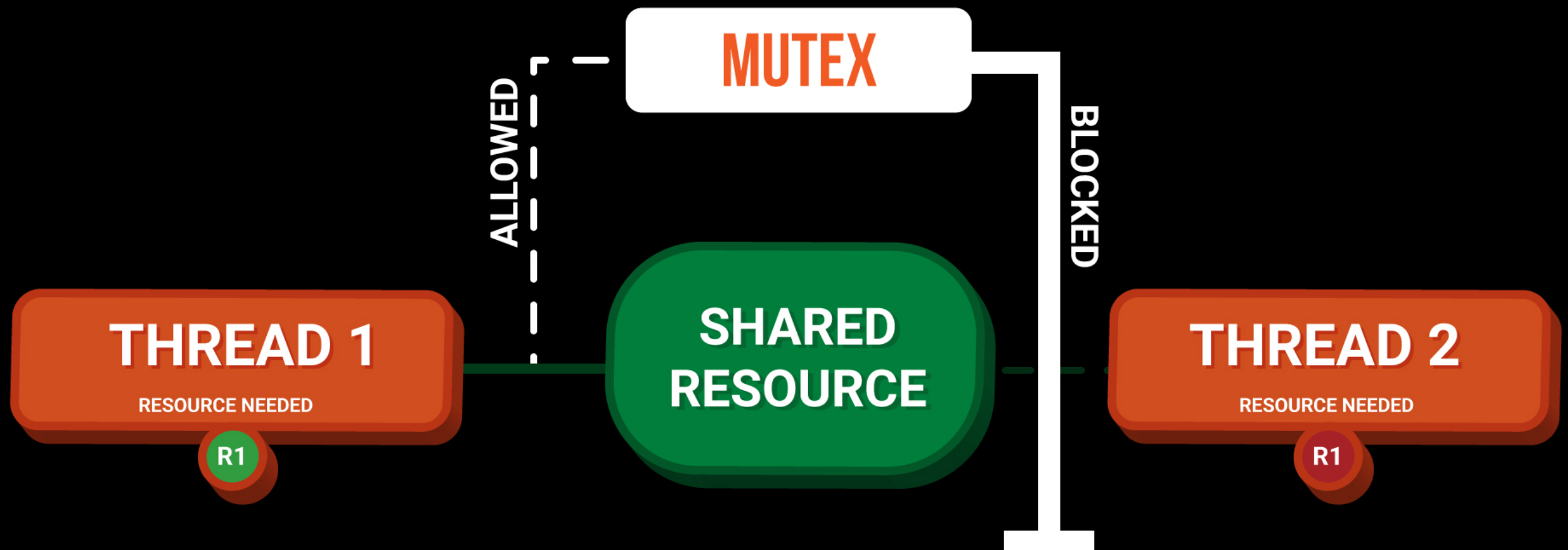
TERMINOLOGY FIRST

DEADLOCK

RESOURCE 1

ASSIGNED

WAITING...

PROCESS 1

RESOURCES NEEDED

R1 R2

PROCESS 2

RESOURCES NEEDED

R1 R2

RESOURCE 2

WAITING...

ASSIGNED

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

MUTEX

Mutex (The name comes from "mutual exclusion") is the method of ensuring that only one process can perform a certain operation at a time.

WHAT'S THE PROBLEM: DINING PHILOSOPHERS

In the *dining philosophers problem*, N philosophers sit at a table. In front of each is a plate of spaghetti. Between each pair of adjacent philosophers is a fork. The philosophers *use a two-handed approach* to eating spaghetti, so each needs two forks to eat. The philosophers' goal is to *eat, put down both forks for a while to think, and eat again*. They repeat this process until they have fathomed all the mysteries of the universe. To make the problem harder, the philosophers are not allowed to talk to each other.

WHAT IS THE ALGORITHM ?

WHAT IS THE ALGORITHM ?

Repeat forever:
    Think until the left fork is available. Pick it up.
    Think until the right fork is available. Pick it up.
    Eat until full.
    Put down the left fork.
    Put down the right fork.
    Think until hungry.

Unfortunately, this algorithm can lead to a **deadlock**. Suppose the philosophers are all quite similar, and they all start the algorithm at the same time. Initially every philosopher needs that the fork on his left is available, so each picks up his left fork. At this point, every fork has been picked up by the philosopher to its right, so every philosopher is stuck waiting for the fork on his right.

SIMULATIONS

Possible Outcomes:

1.
2.
3.

Resource Starvation
Deadlock
Success

HOW TO PREVENT THEN ?

There are couple ways to solve this problem:

– Randomization

– Resource Hierarchy

   – Djikstra's solution, Smallest numbered fork first

– Waiter

   – May result reduced parallelism. Waiting for the response even though their forks are available

– Chandra/Misra

THANKS FOR LISTENING!
QUESTIONS ?

REFERENCES

- https://github.com/NAUniversity/AlgorithmsBook/blob/master/Chapter%2018%20-%20Distributed%20Algorithms.ipynb

- https://en.wikipedia.org/wiki/Dining_philosophers_problem

- https://www.youtube.com/watch?v=p0SKPpC5r9U