What is a thread?

Answer: A thread is a road!

(Start) _____ (End)

The road is narrow, so only one car can travel on it at a time:

(Start) 🚗_____ (End)

But...there can be lots of roads, so many cars can travel at the same time!

Car 1

(Start) (End)

Car 2

(Start) (End)

Car 3

(Start) (End)
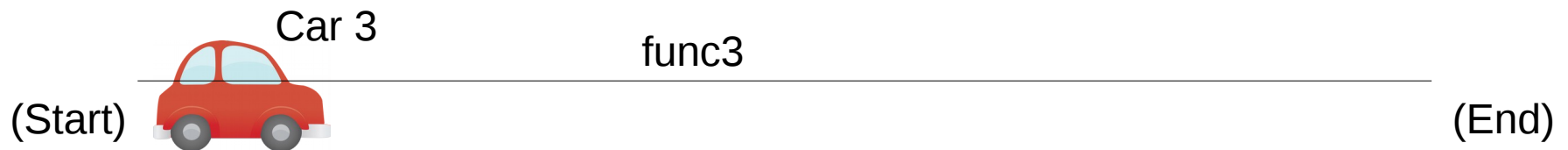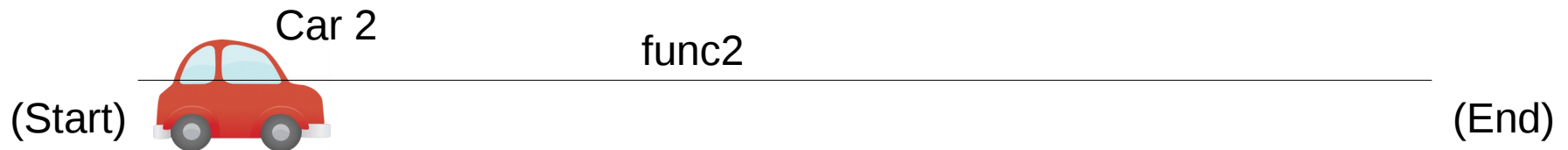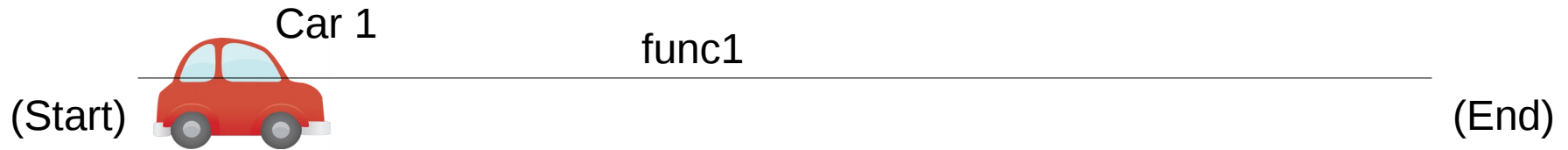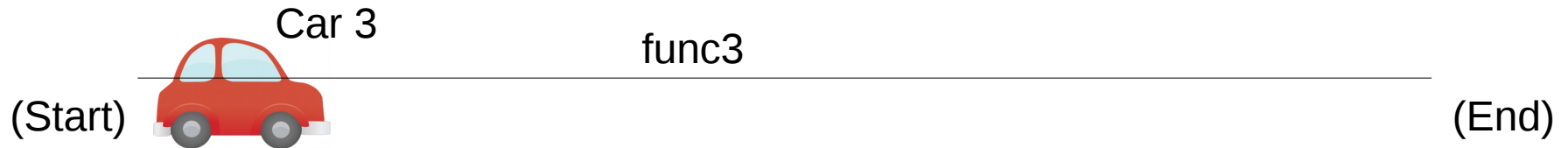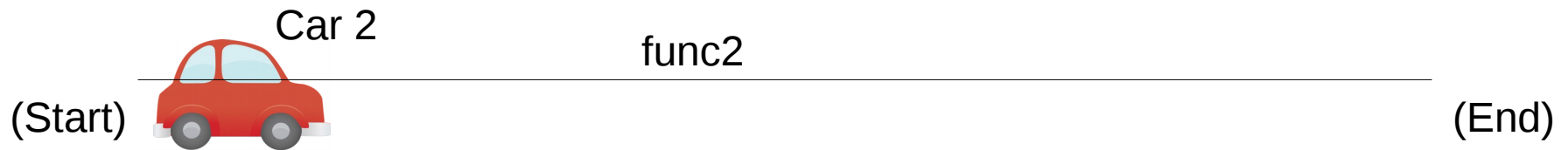
What are the names of the roads, you ask?
Look at the code in threads-basic.c:

Car 1
func1

(Start)   (End)

Car 2
func2

(Start)   (End)

Car 3
func3

(Start)   (End)

The position of the car tells you how far along the road it is,
i.e., which line of code will be executed next.

What's inside each car?
The argument passed to the thread (void * argp) and
any local variables created along the way

Car 1

func1

(Start)   (End)

Car 2

func2

(Start)   (End)

Car 3

func3

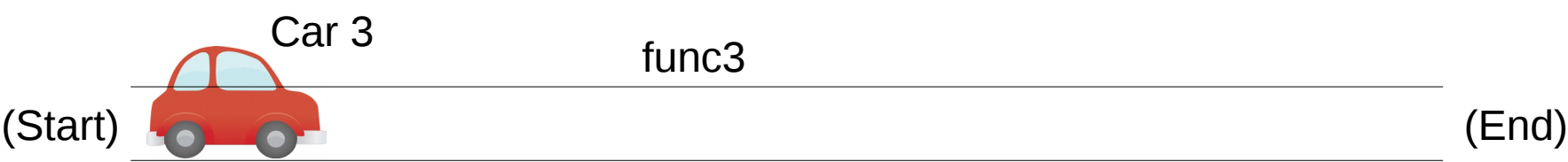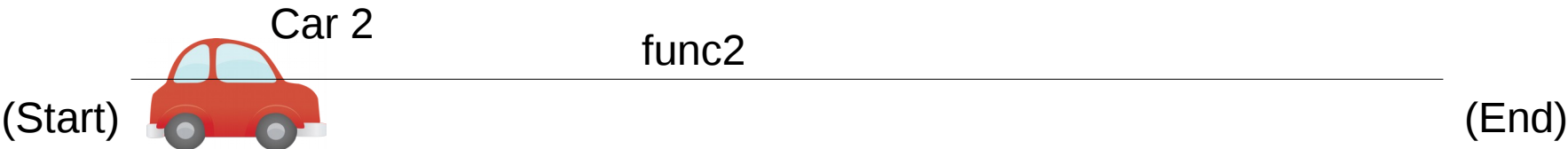(Start)   (End)
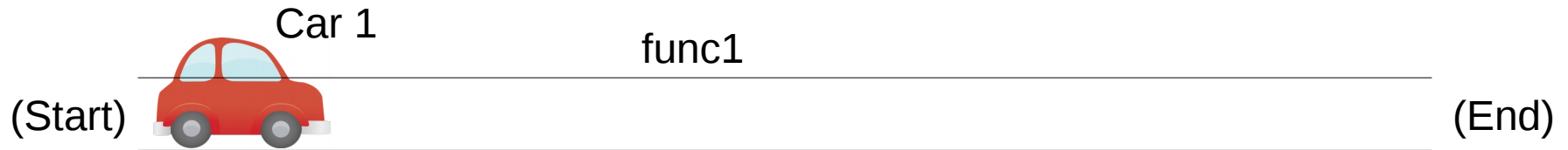
But wait...there's one more thread: main (where it all starts)!

Car 0

main

(Start) (End)

Car 1

func1

(Start) (End)

Car 2

func2

(Start) (End)

Car 3

func3

(Start) (End)

So, technically:

Car 0

main

(Start)   (End)

pthread_create

Car 1

func1

(Start)   (End)

pthread_create

Car 2

func2

(Start)   (End)