More on Semaphores

Sarah Diesburg

Operating Systems

CS 3430

The Pigeon Network Scenario

- Pigeons are good message carriers
  - Reasonably reliable
  - Relatively fast for less developed rural areas
  - Can sense magnetic field lines

Here is the Story…

- There are two towns—Mars and Venus
  - Mars has all male pigeons
  - Venus has all female pigeons
- Each town delivers messages to the other
  - By sending a pigeon through the shared flying path
  - And waiting for the same pigeon to fly back as an acknowledgement

Here is the Story…

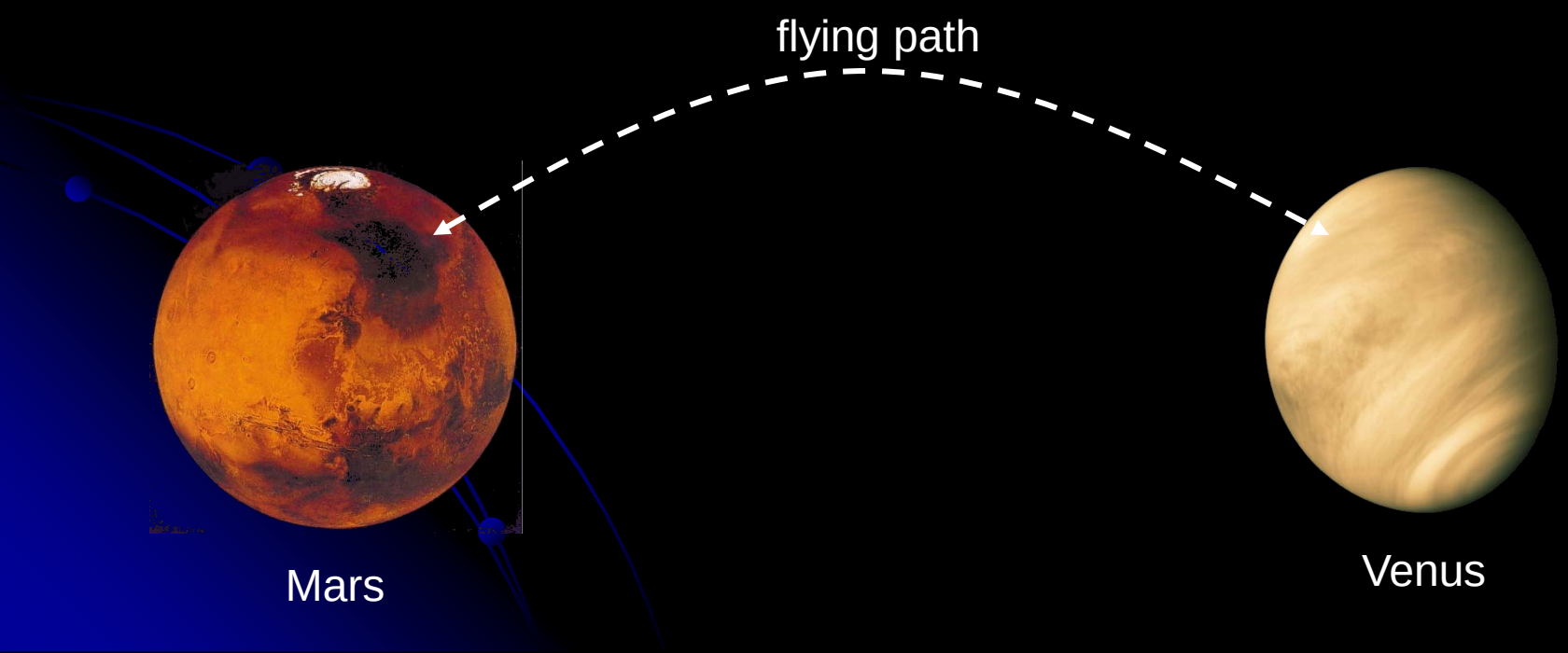
- Based on experience
  - Whenever both towns send messages simultaneously, the reliability drops significantly
  - Pigeons of opposite genders decide to take excursions
- Goals of a pigeon network:
  - Efficiency
  - Fairness

Developing the Solution

- First we need to identify the components of the problem
- Second we develop the simplest solution
  - Then optimize

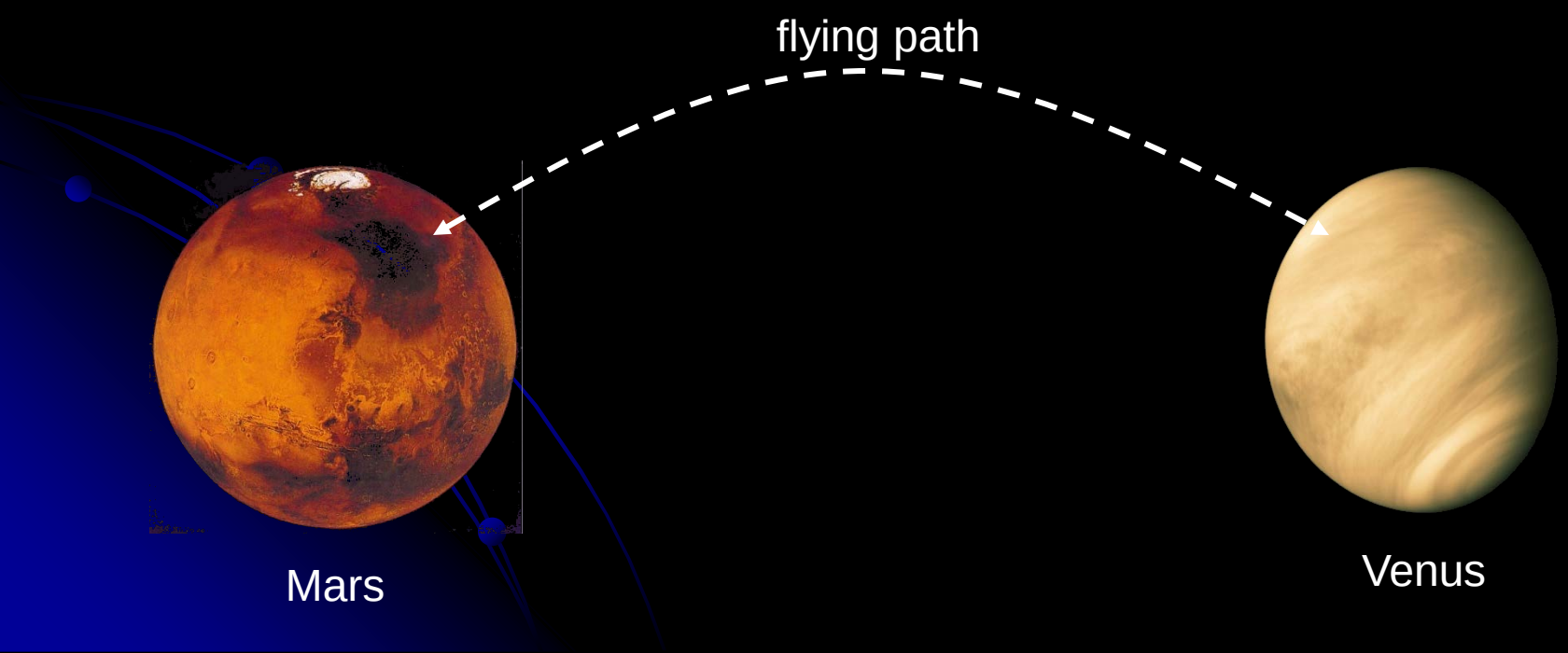
Step 1: Visualization

- Identify
  - Shared resources
  - Scope of shared resources



Step 1: Visualization

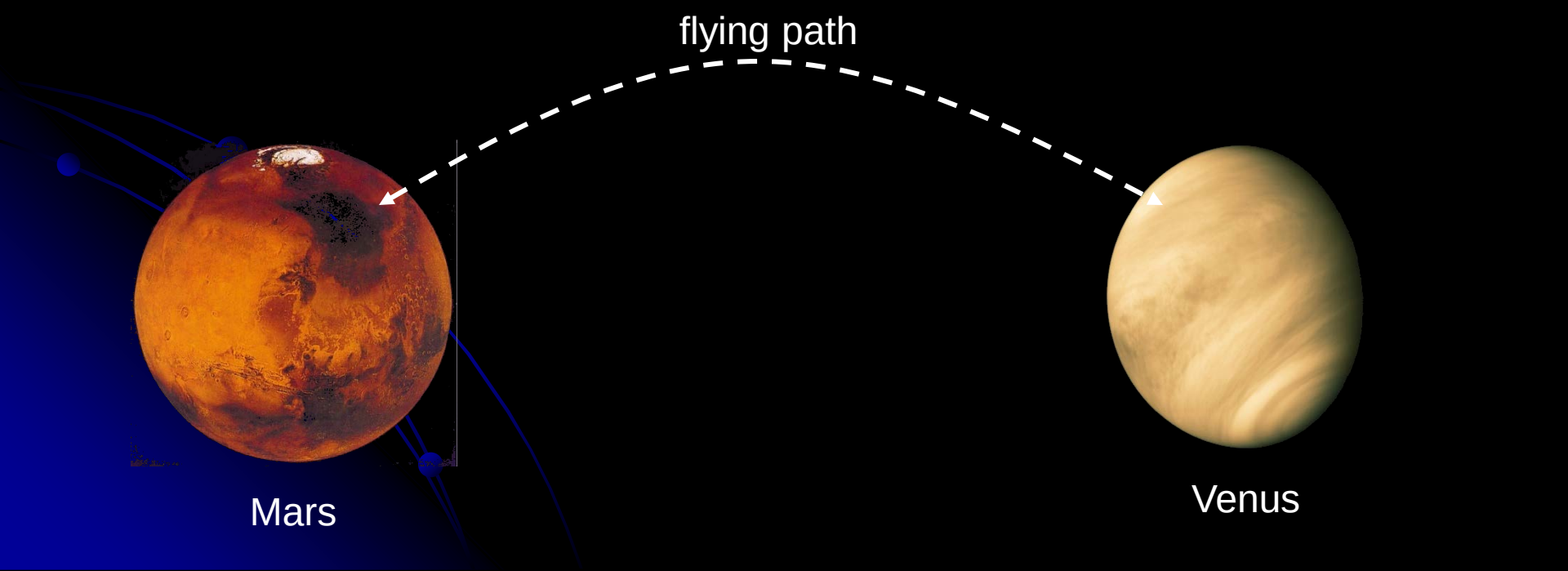
- Identify
  - Shared resource: flying path
  - Scope of the shared resource: global



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

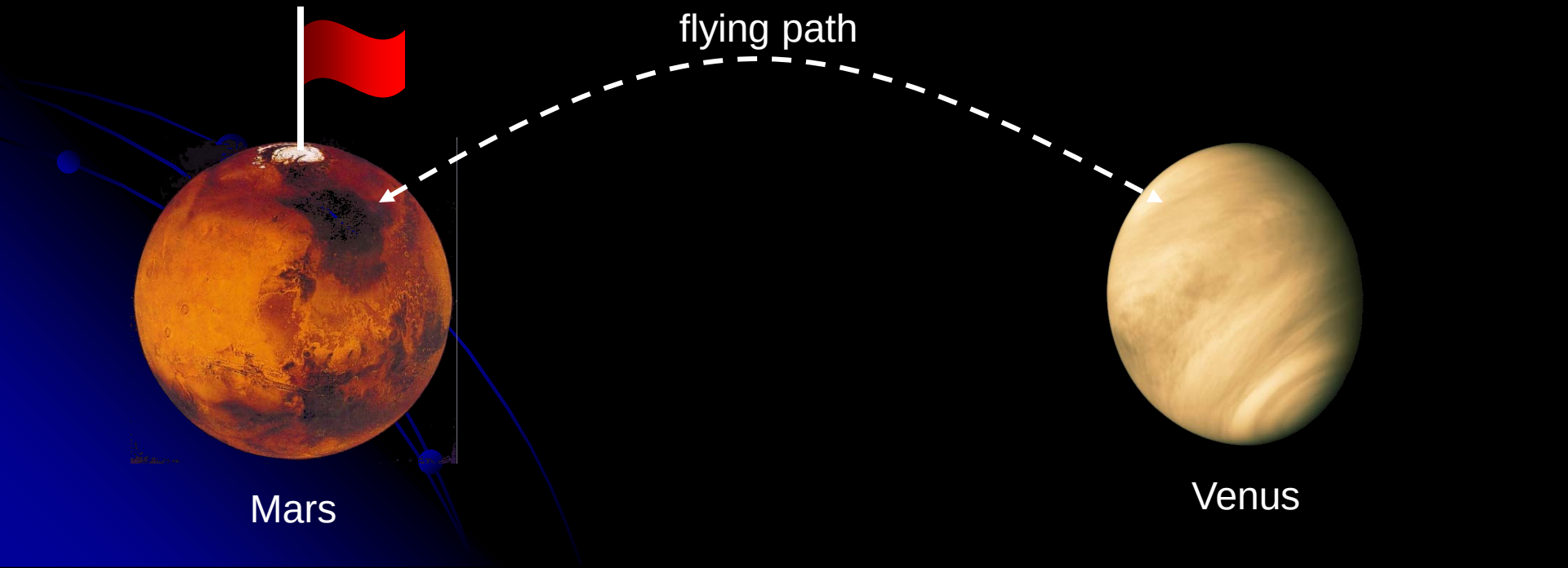
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

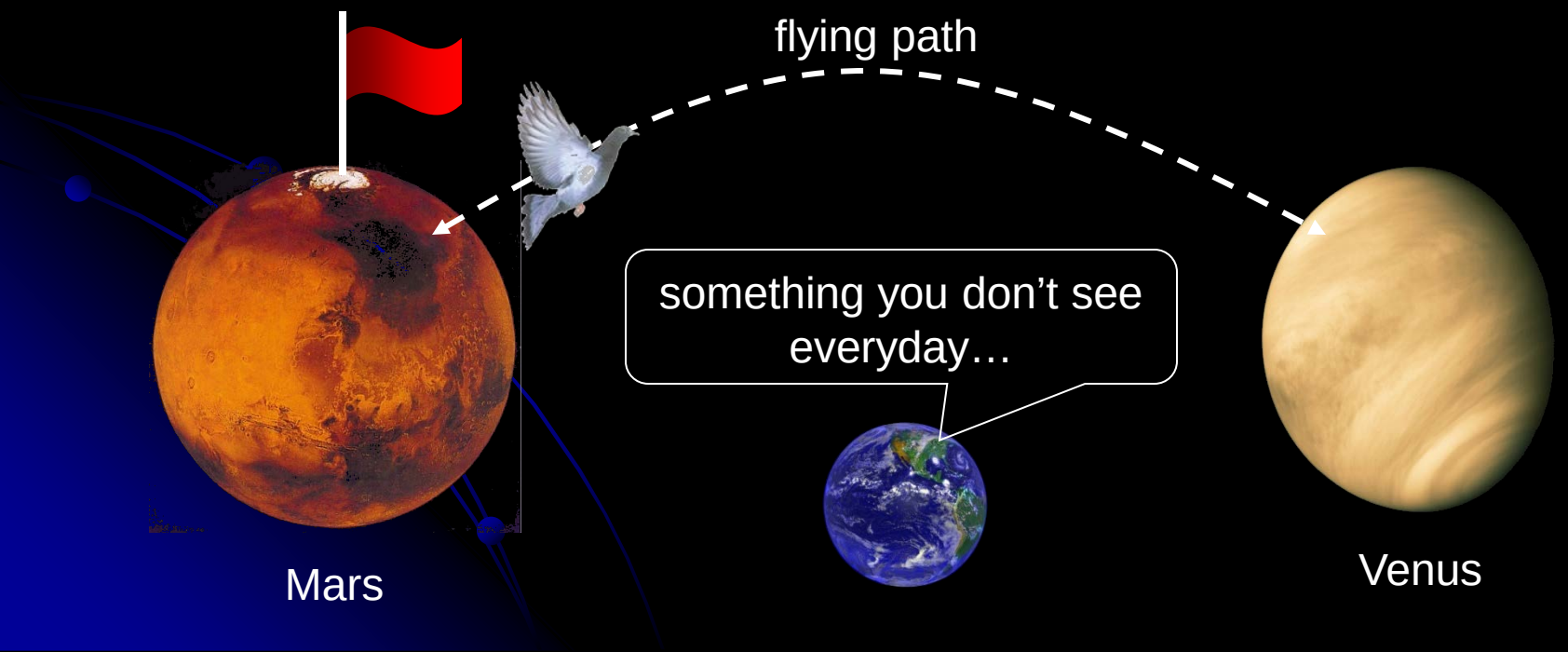
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

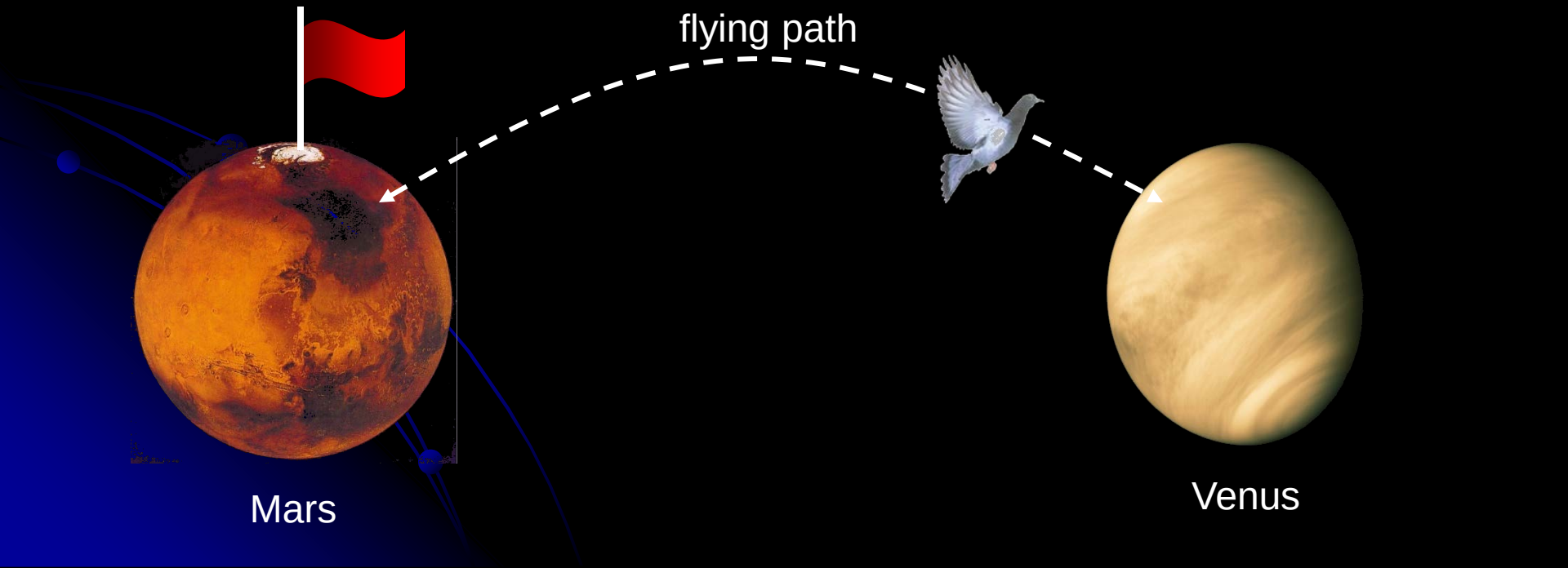
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

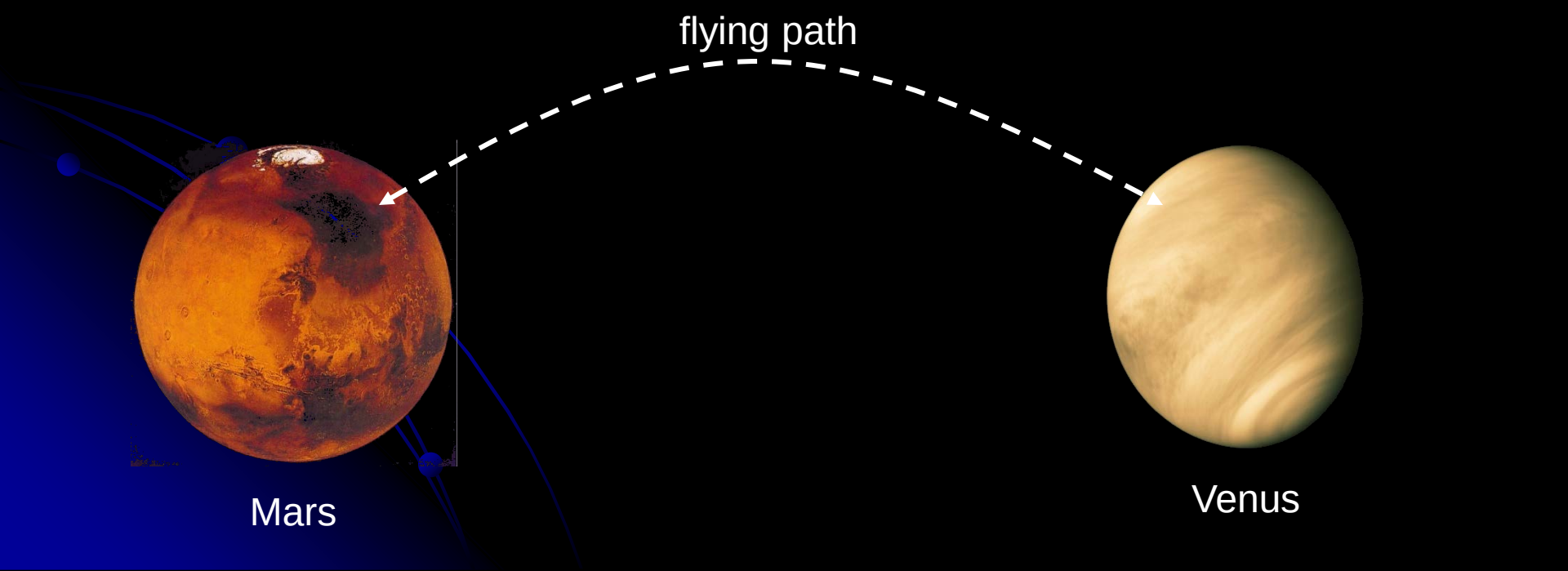
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

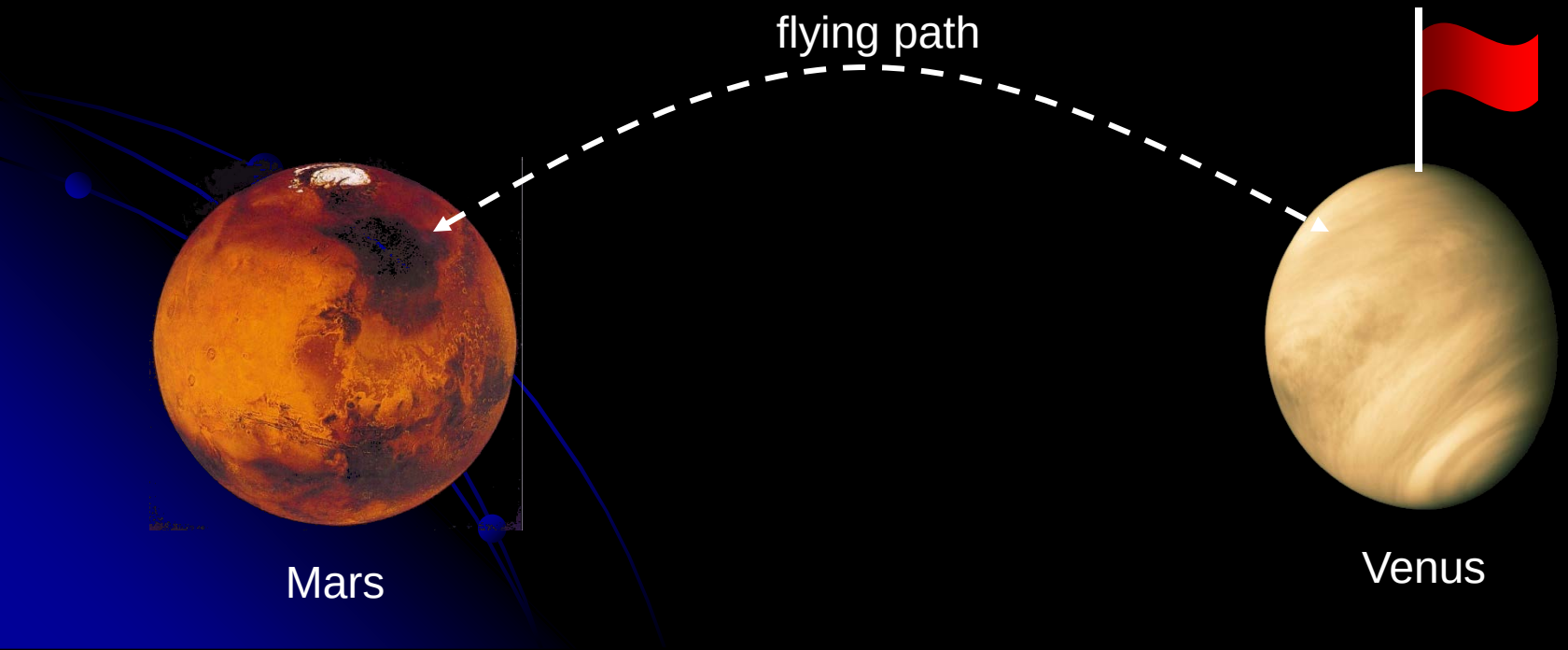
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

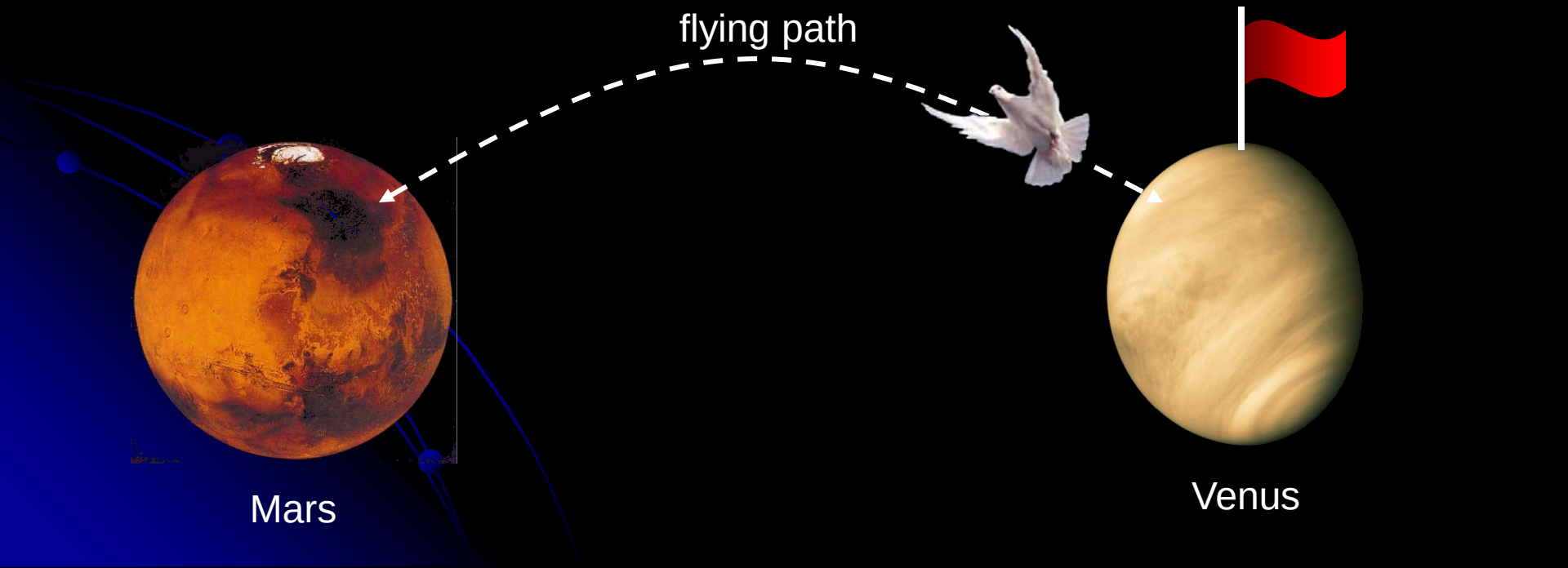
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

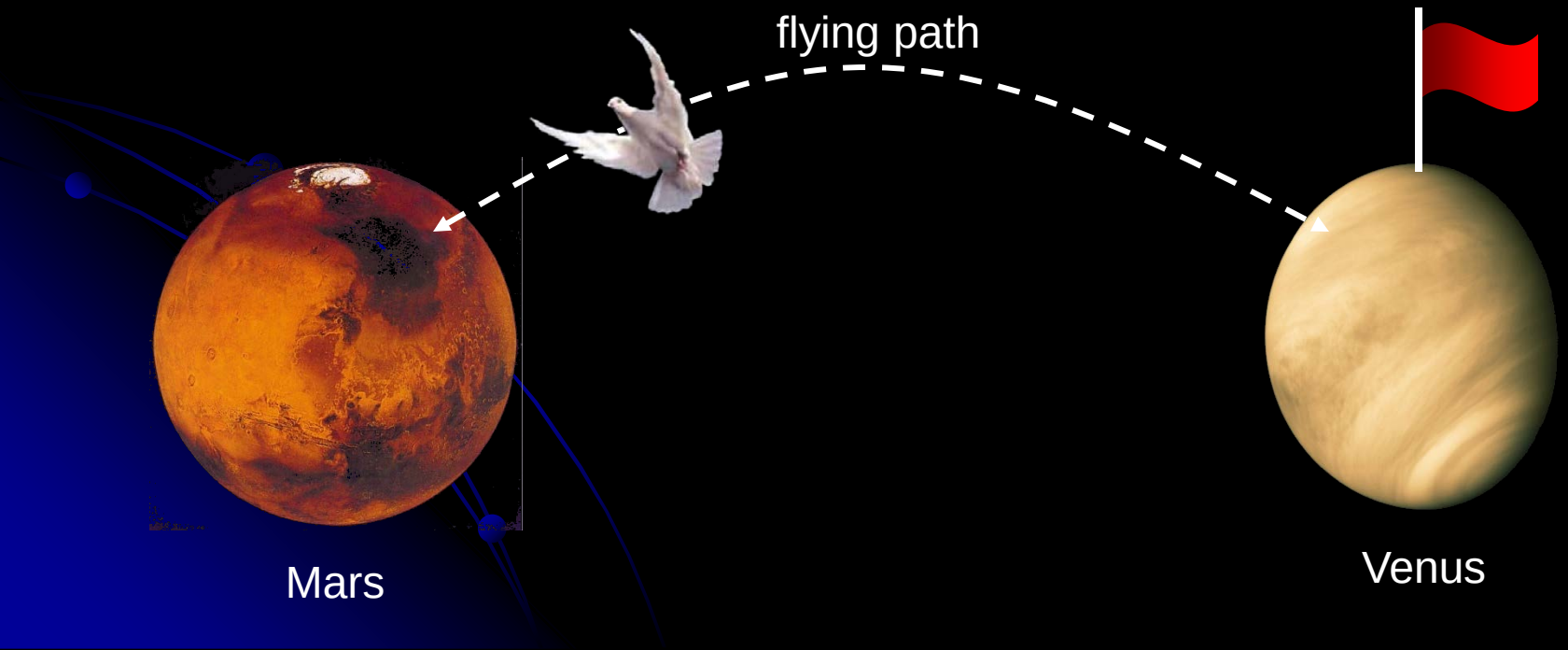
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

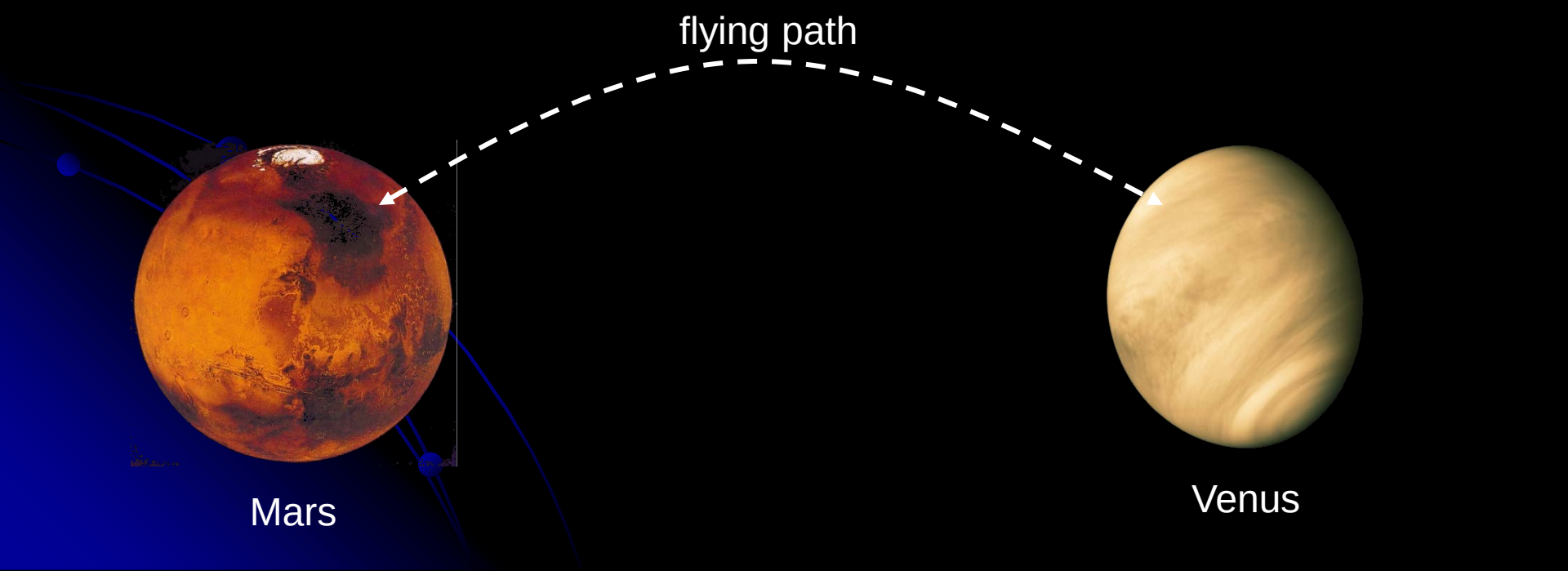
```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

```
P(flyingPath);
// send the message
V(flyingPath);
```



Simplest Implementation

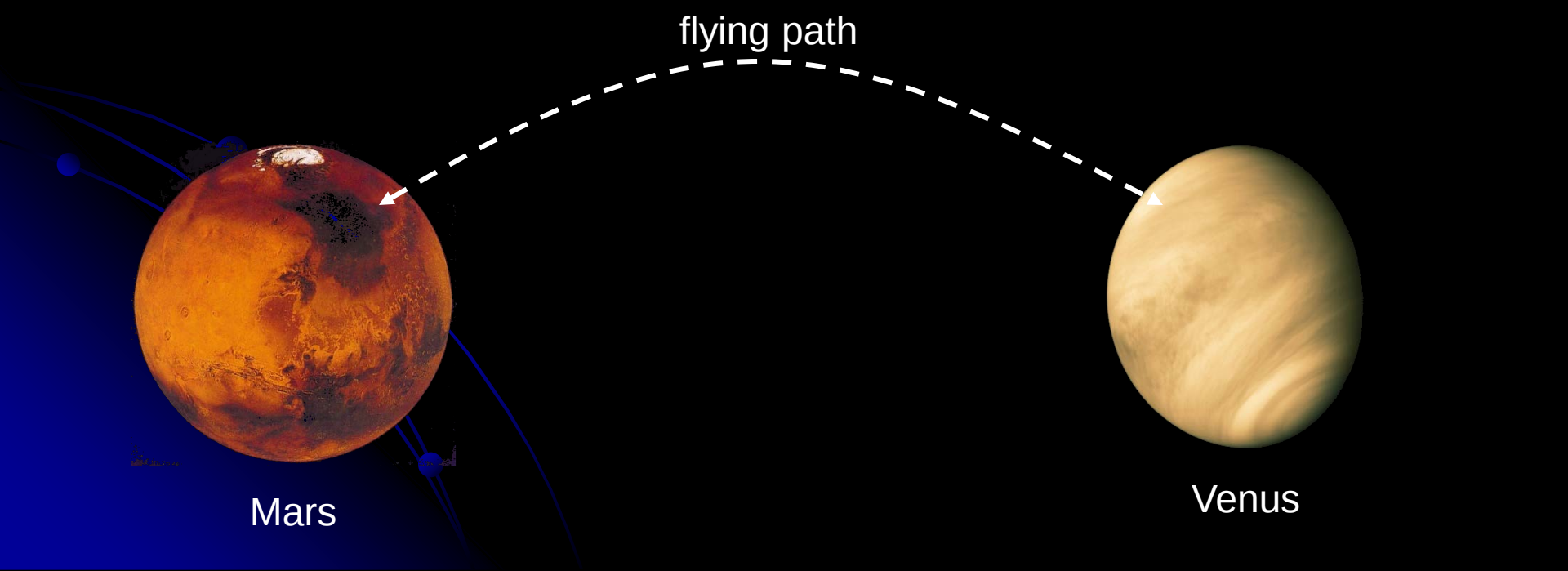
+ Simple

+ Fair

- Not efficient
  - Only one pigeon can fly at a time
  - Venusians love to fly in groups…

Allowing Multiple Venusians in Transit

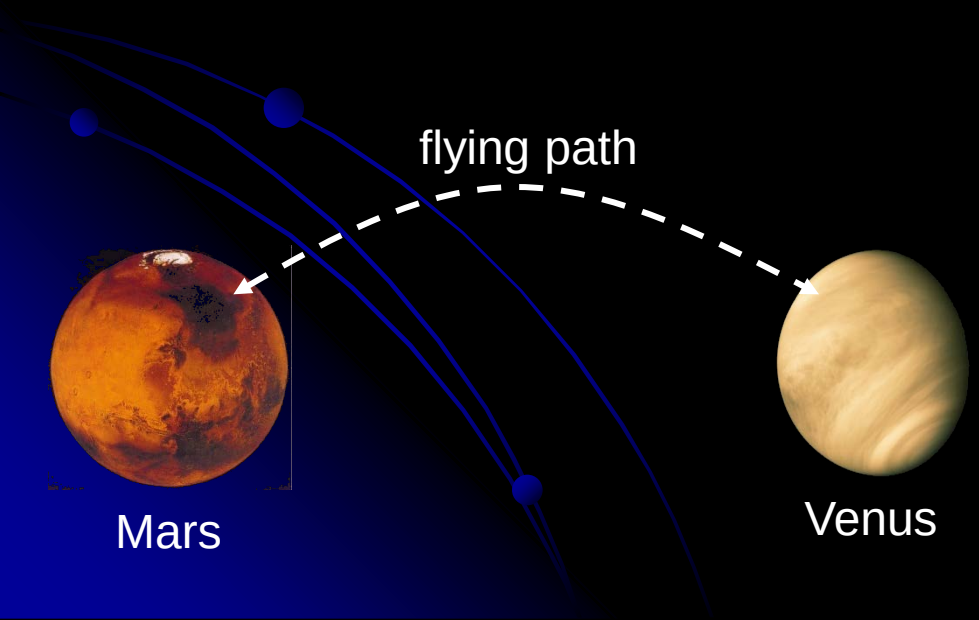
- Resources and scopes: flying path (global), a counter for Venusians in transit (Venus)



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

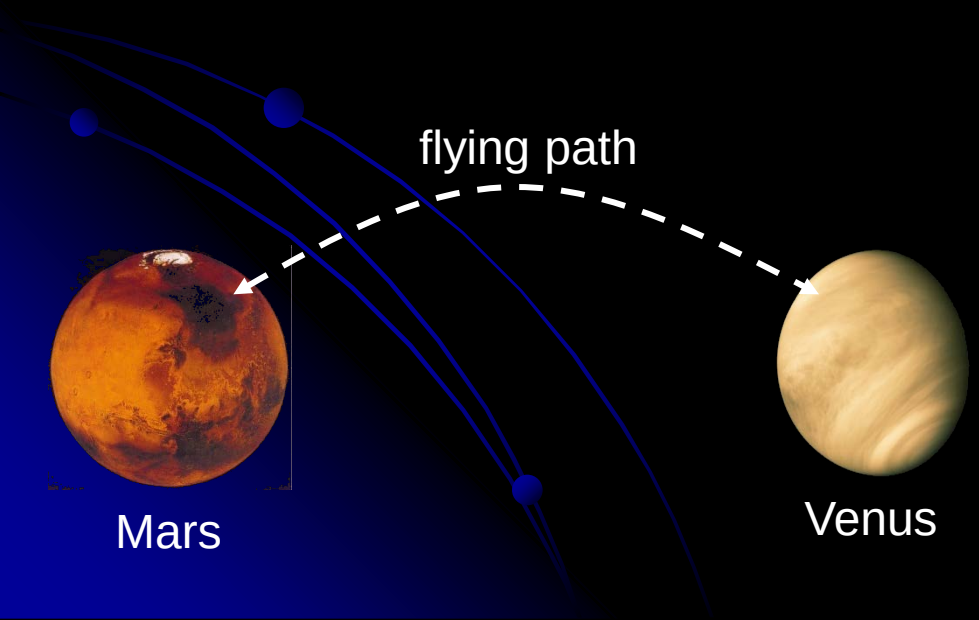
```
int VenusiansInTransit = 0;


++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

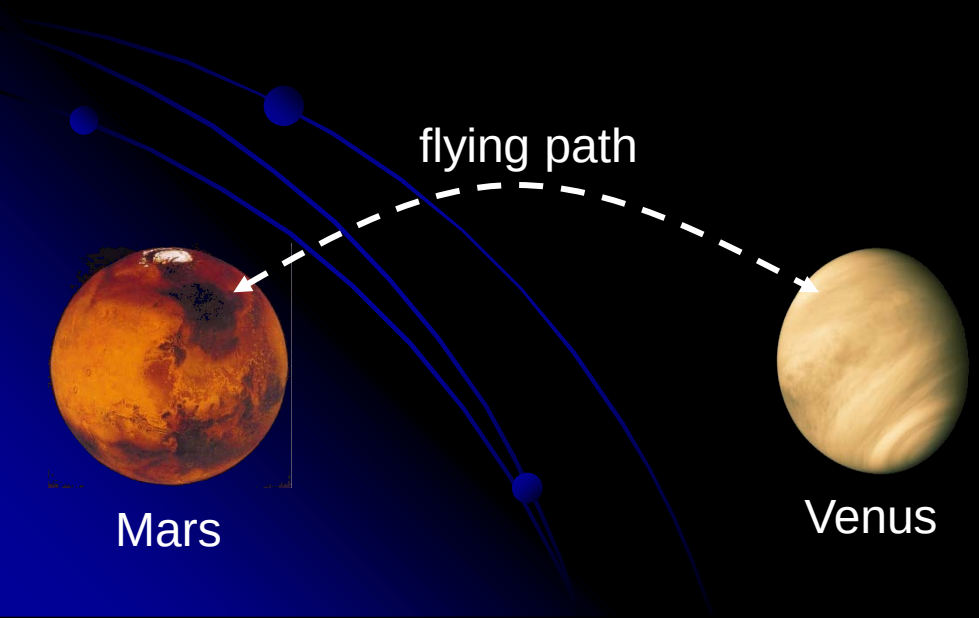
```
int VenusiansInTransit = 0;


++VenusiansInTransit; // == 1
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

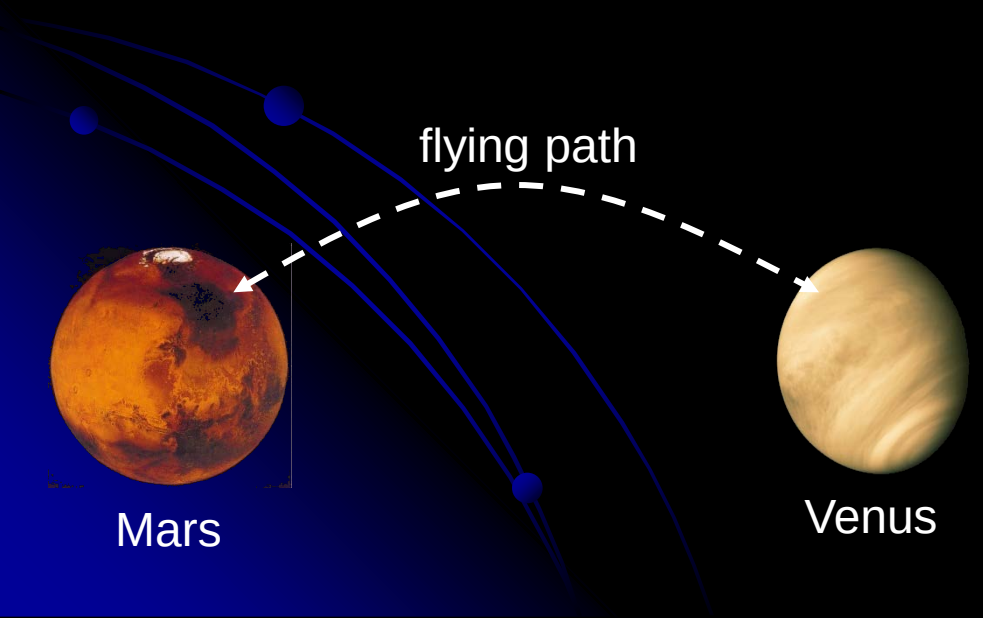
```
int VenusiansInTransit = 0;


++VenusiansInTransit; // == 2
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

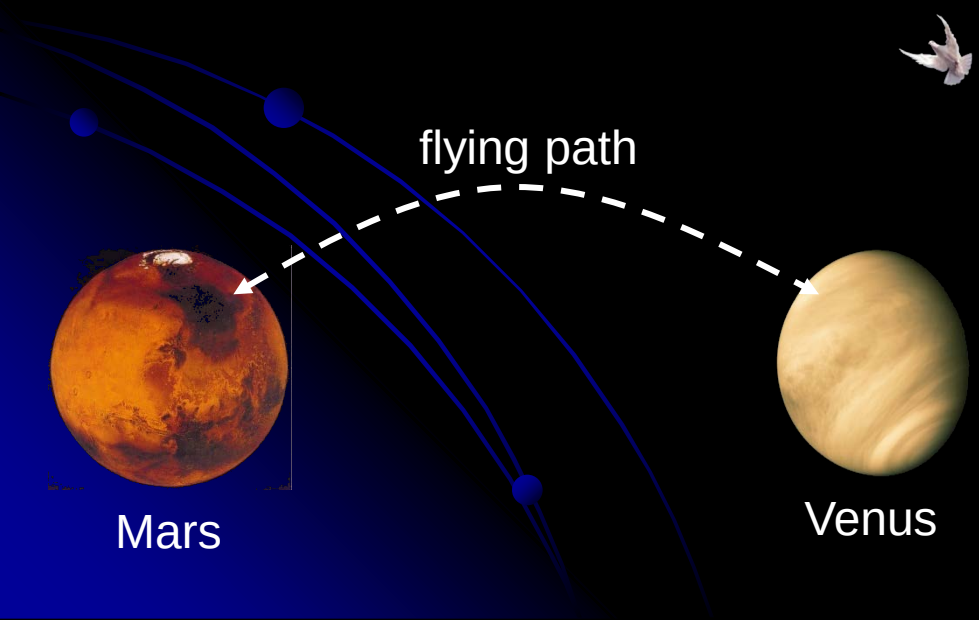
```
int VenusiansInTransit = 0;


++VenusiansInTransit; // == 2
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

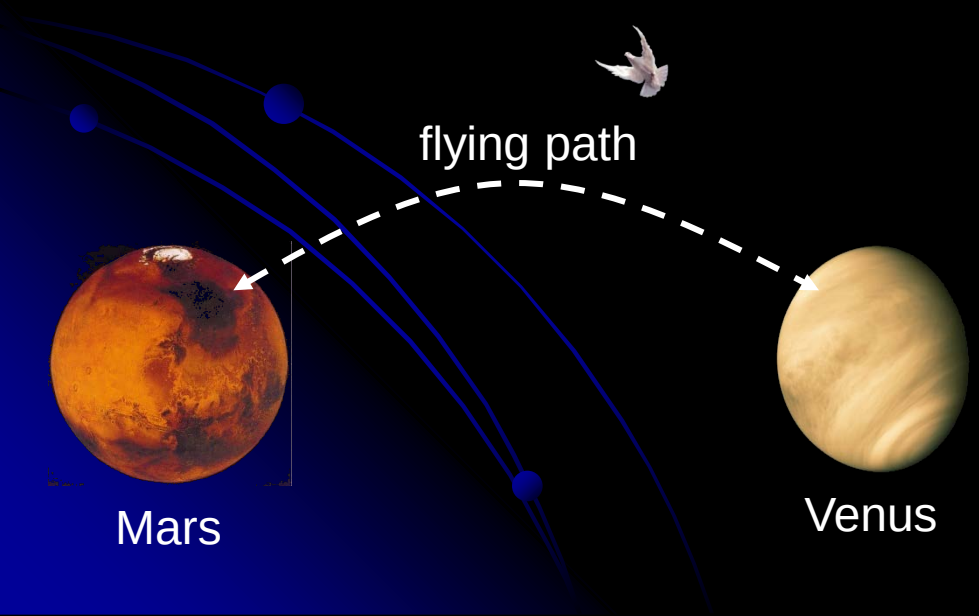
```
int VenusiansInTransit = 0;


++VenusiansInTransit; // == 2
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

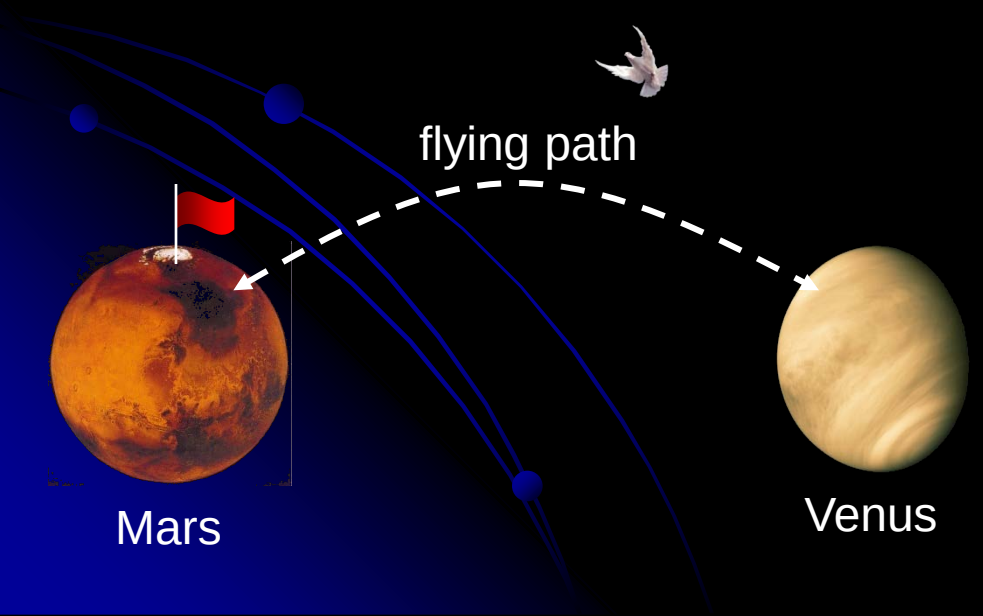
```
int VenusiansInTransit = 0;


++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

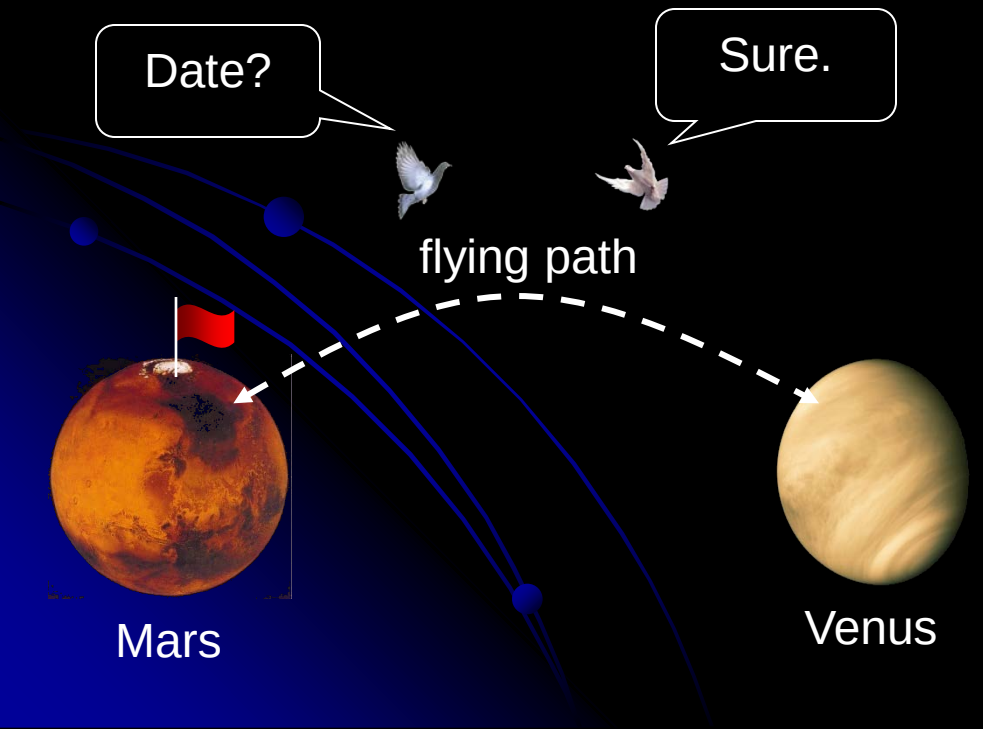
```
int VenusiansInTransit = 0;


++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

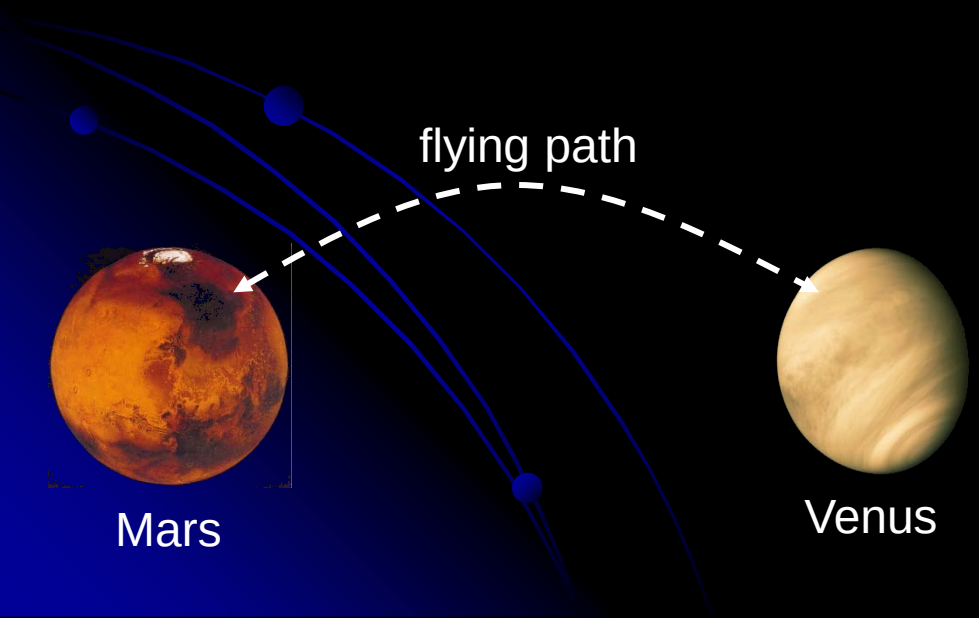
```
int VenusiansInTransit = 0;


++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
```



Allowing Multiple Venusians in Transit Revised

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

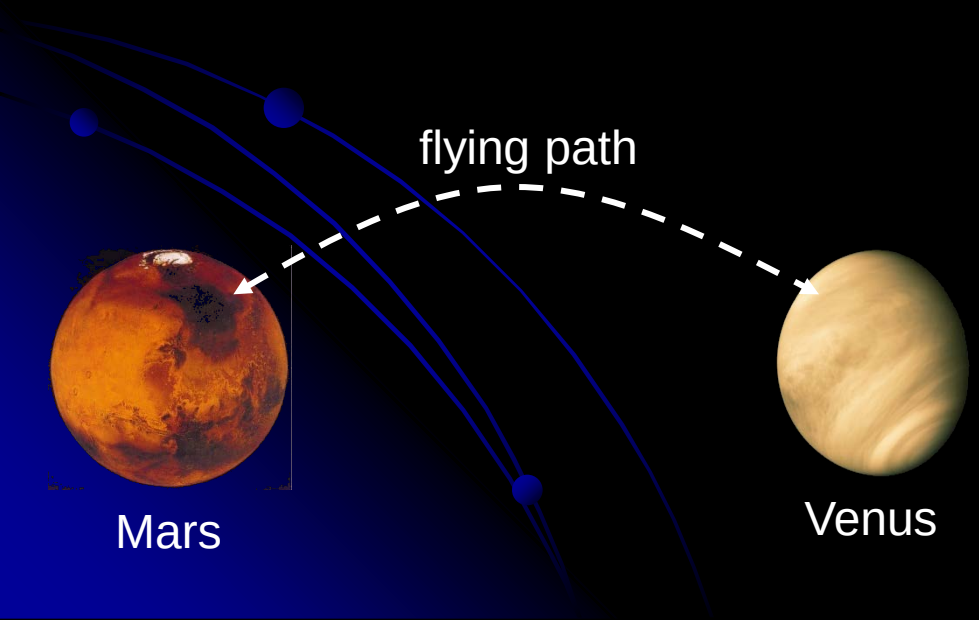
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

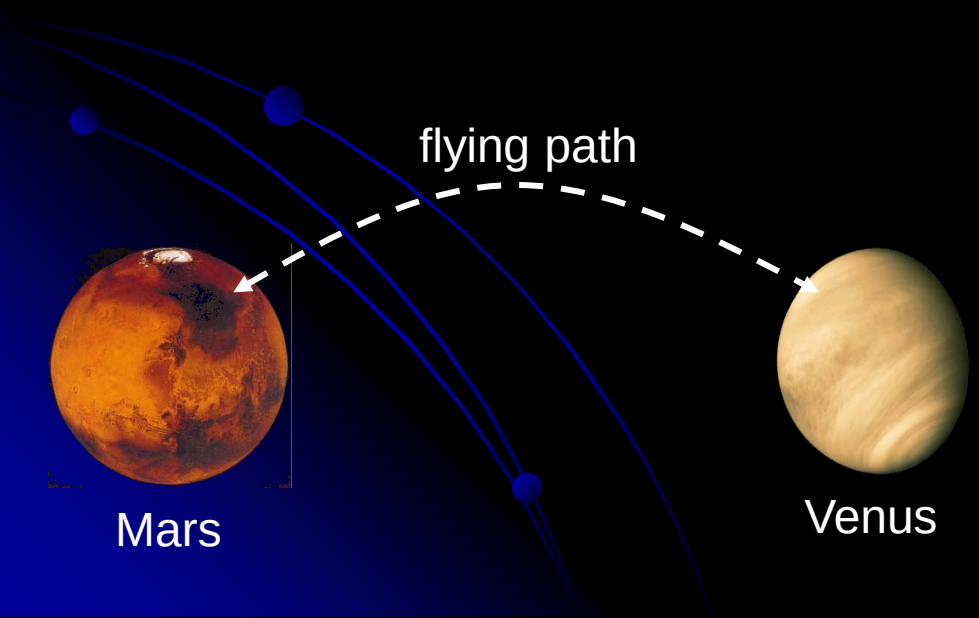
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```

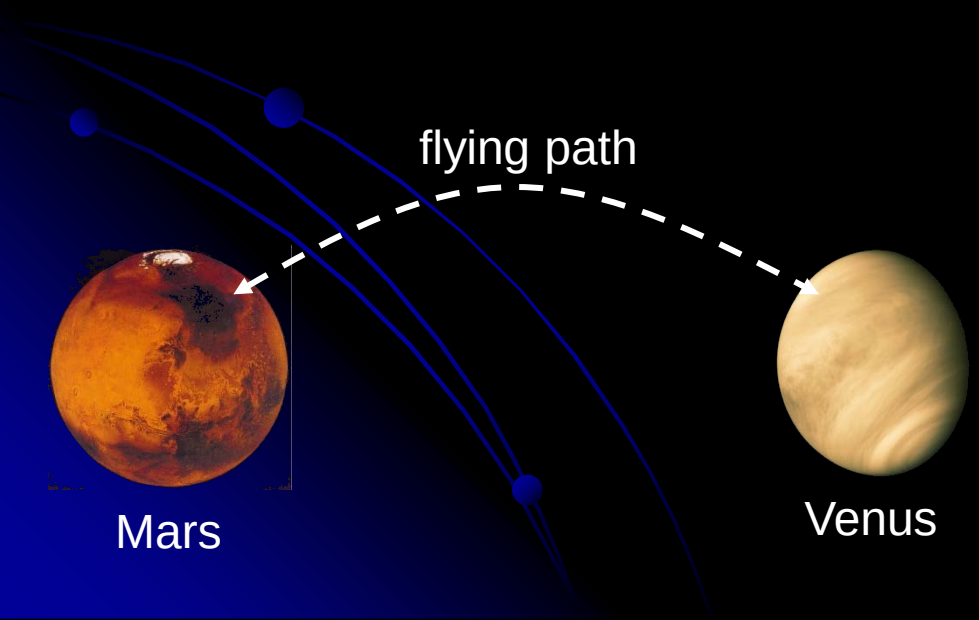


Allowing Multiple Venusians in Transit Revised

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

Wait!

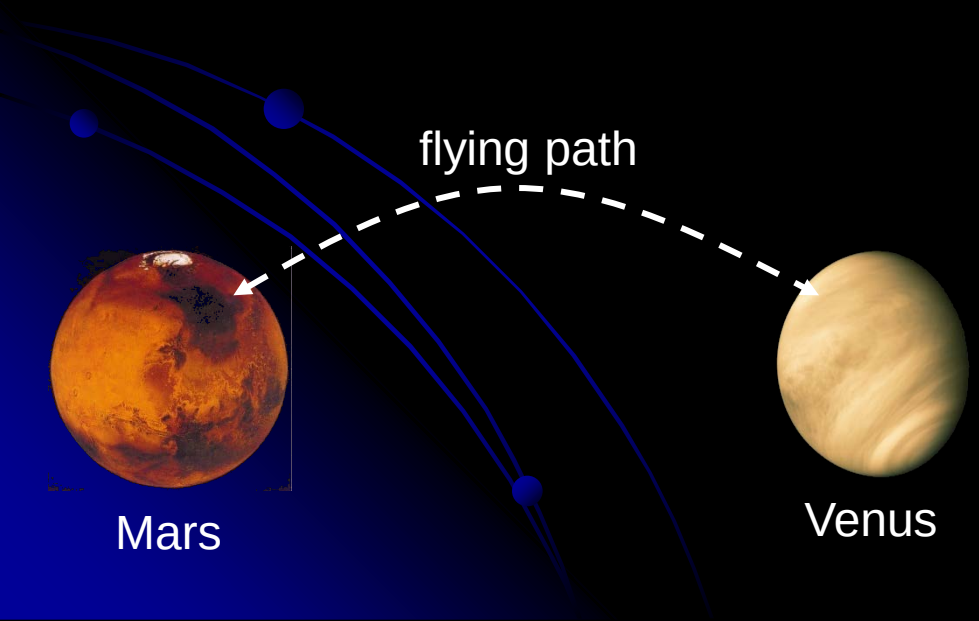
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}

// send the message

--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

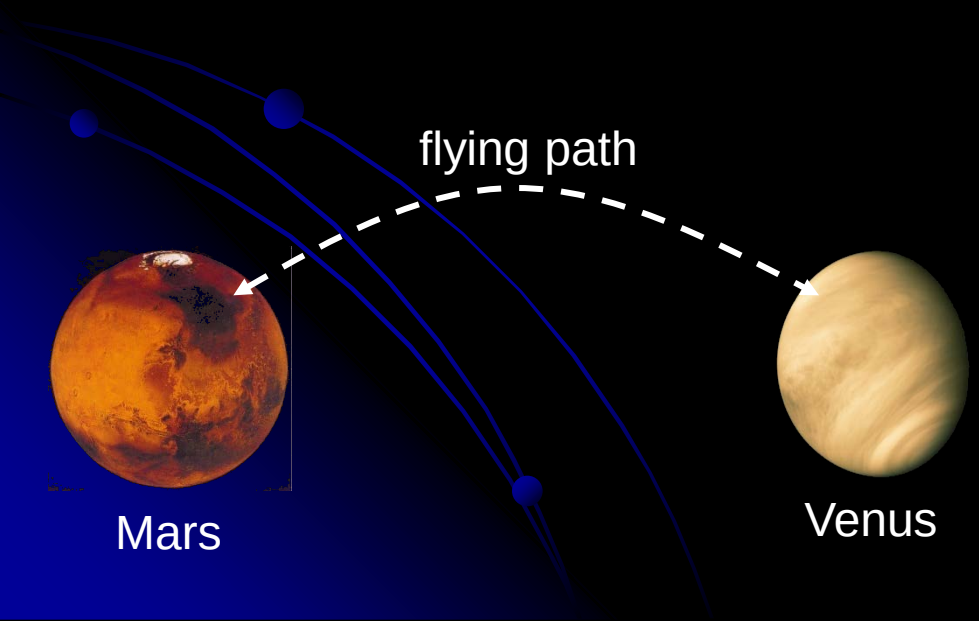
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

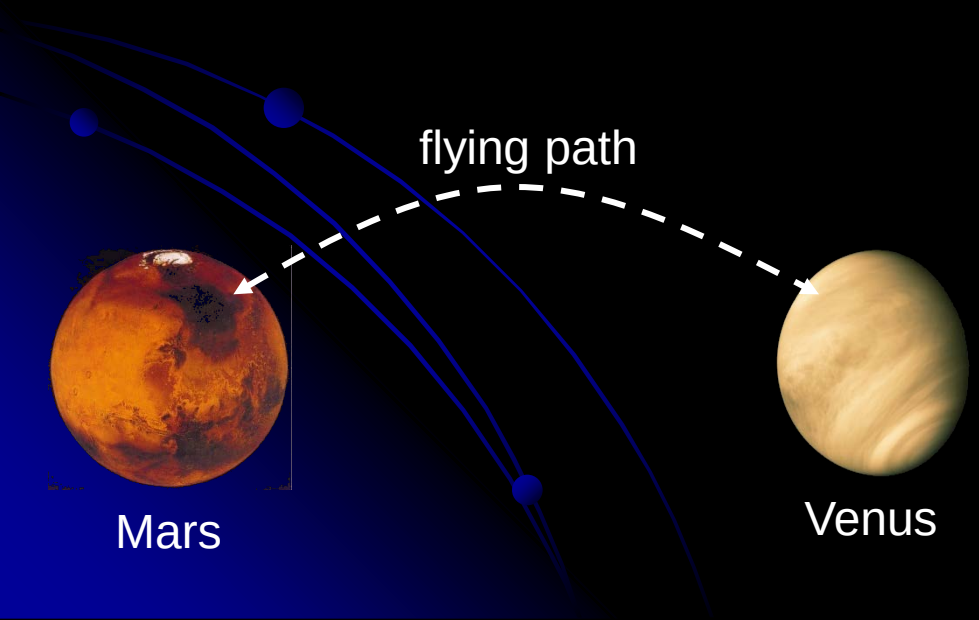
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

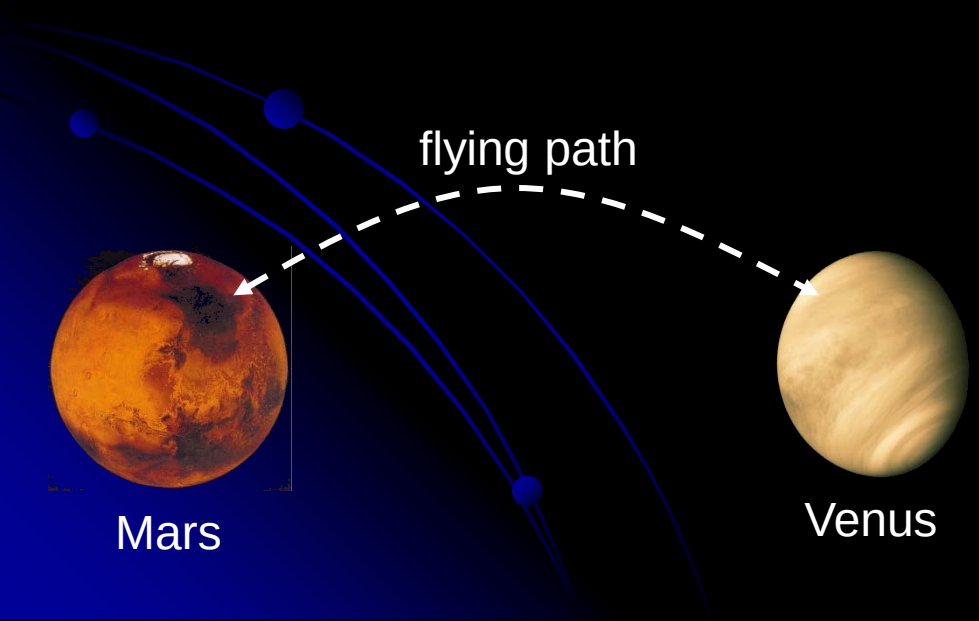
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

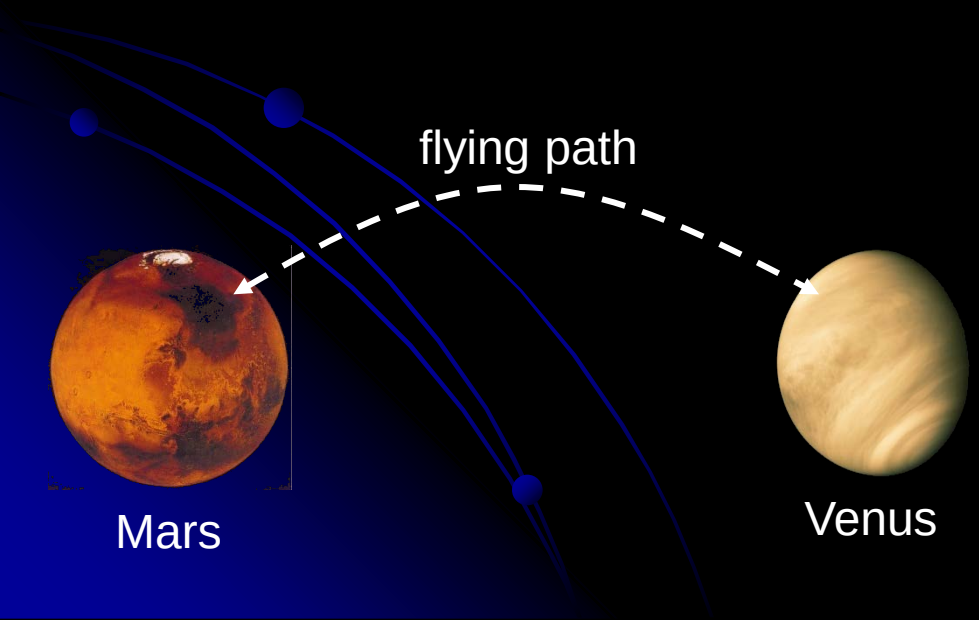
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

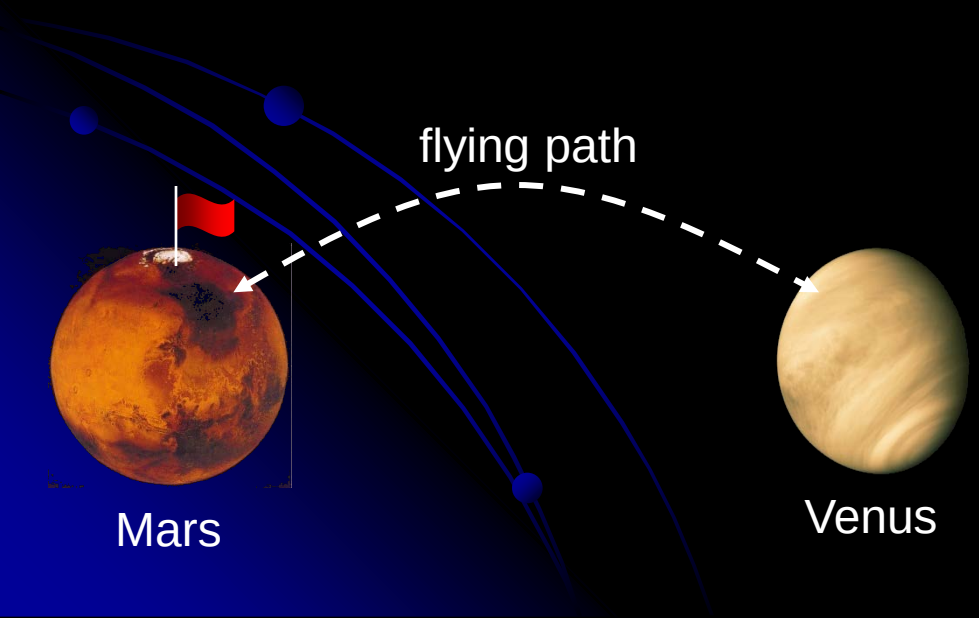
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

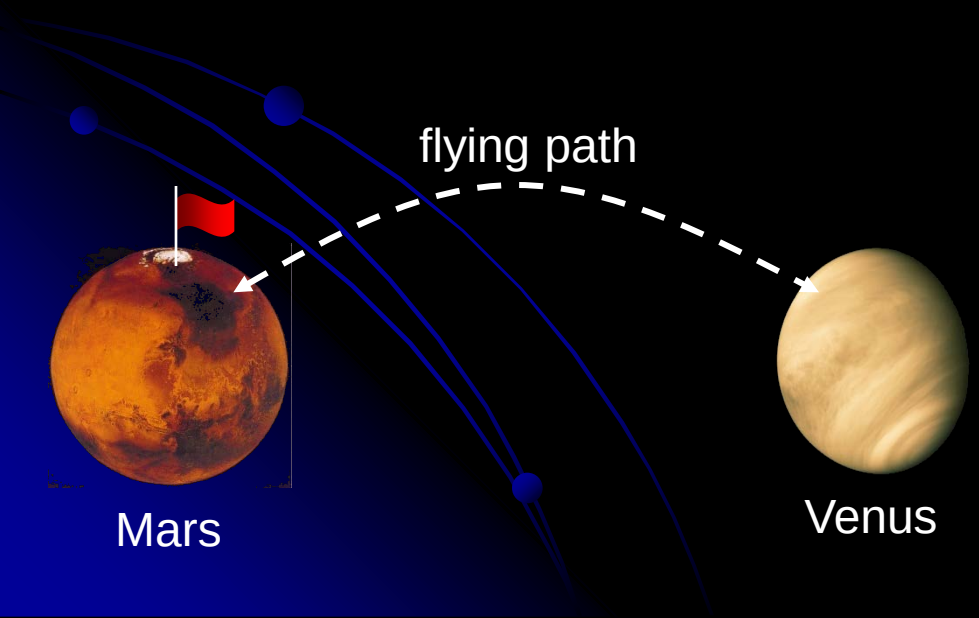
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

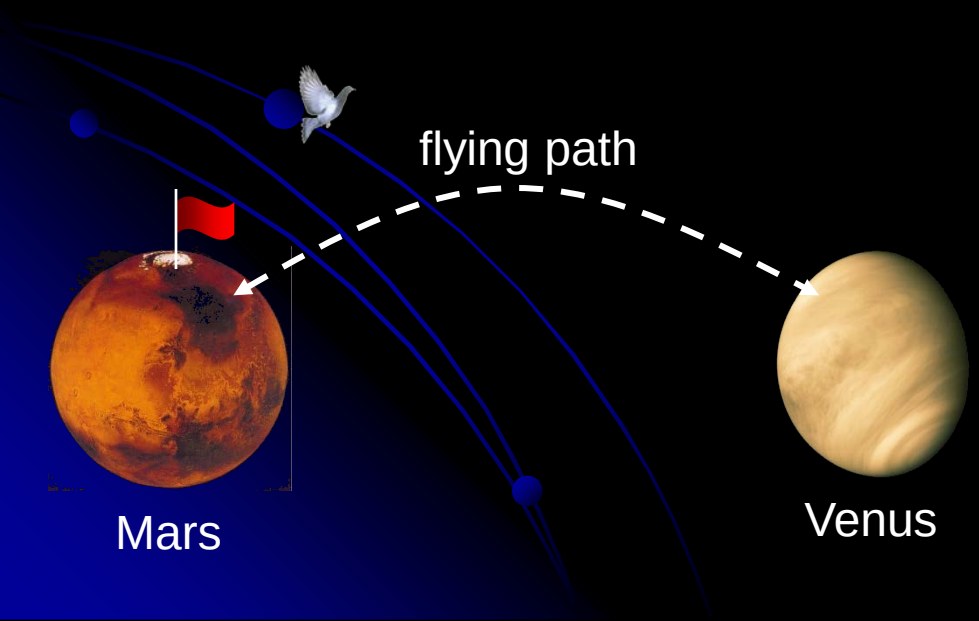
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

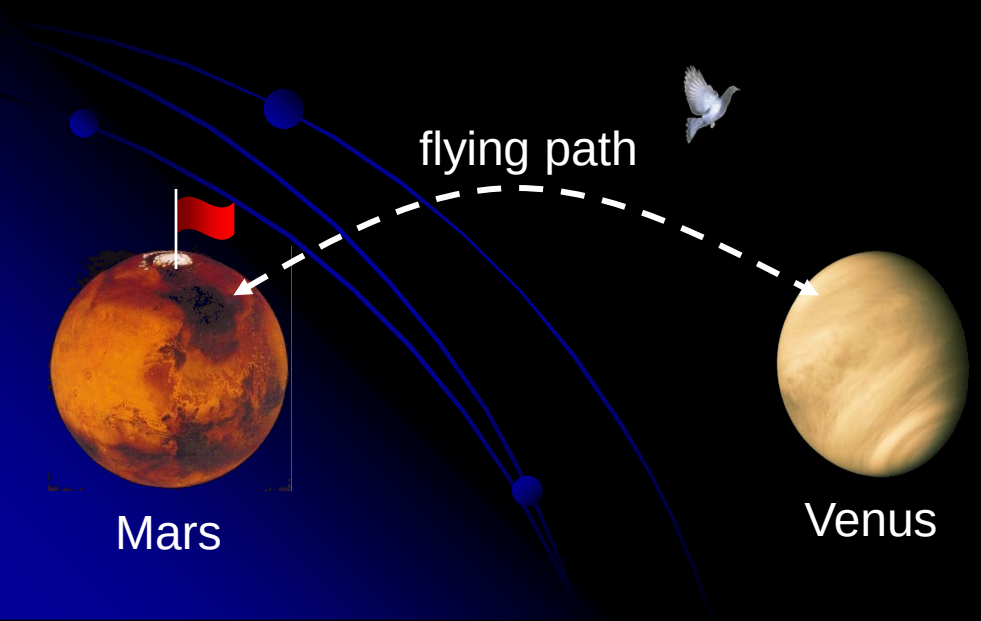
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

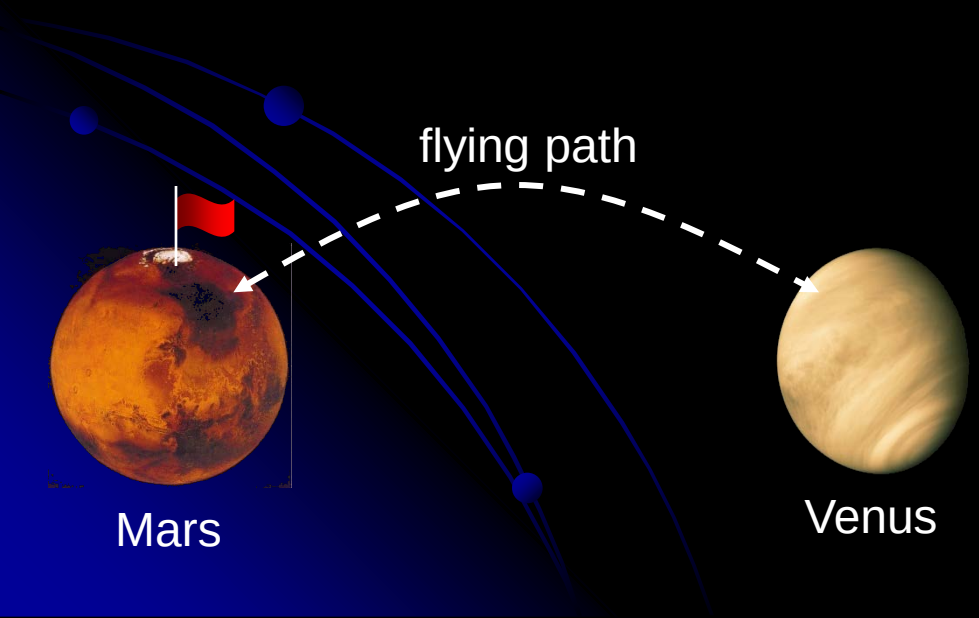
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

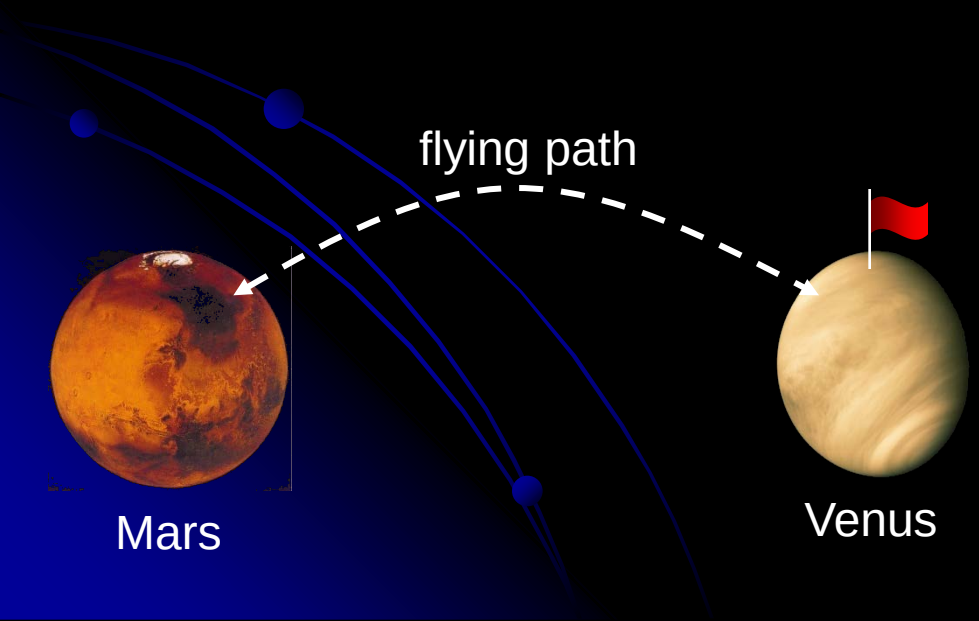
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

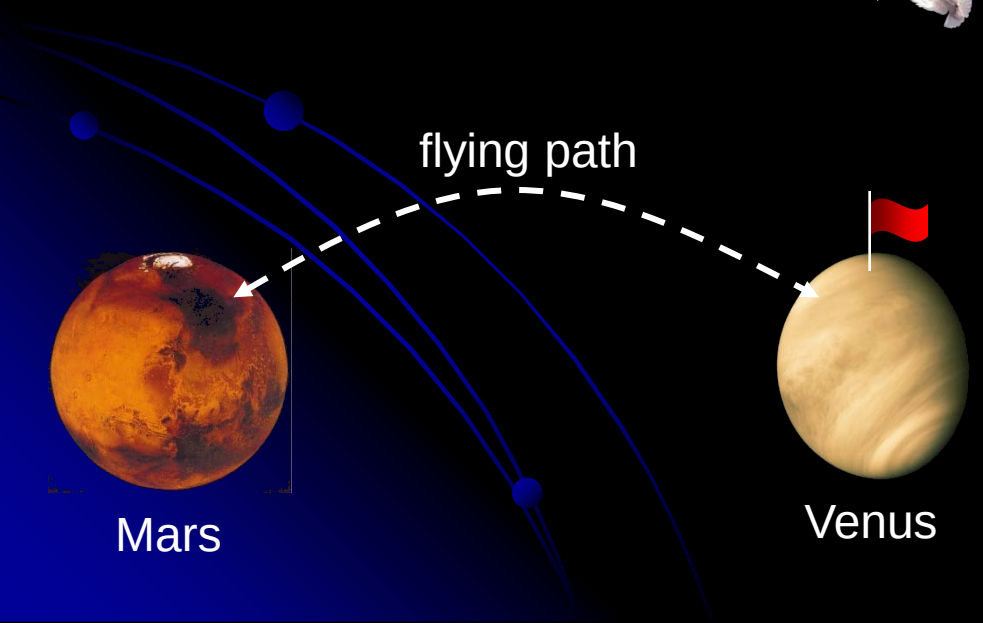
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

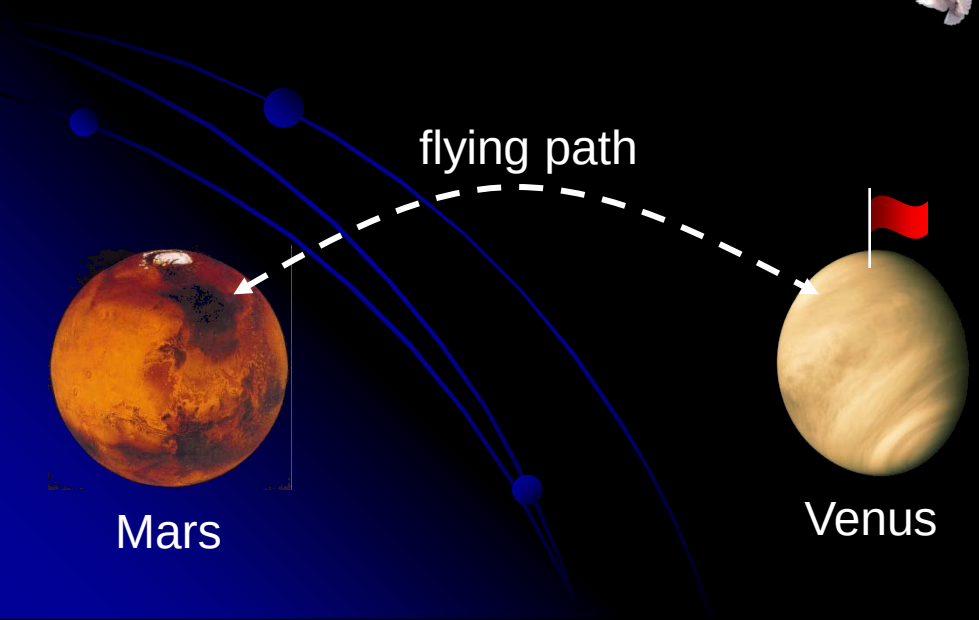
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

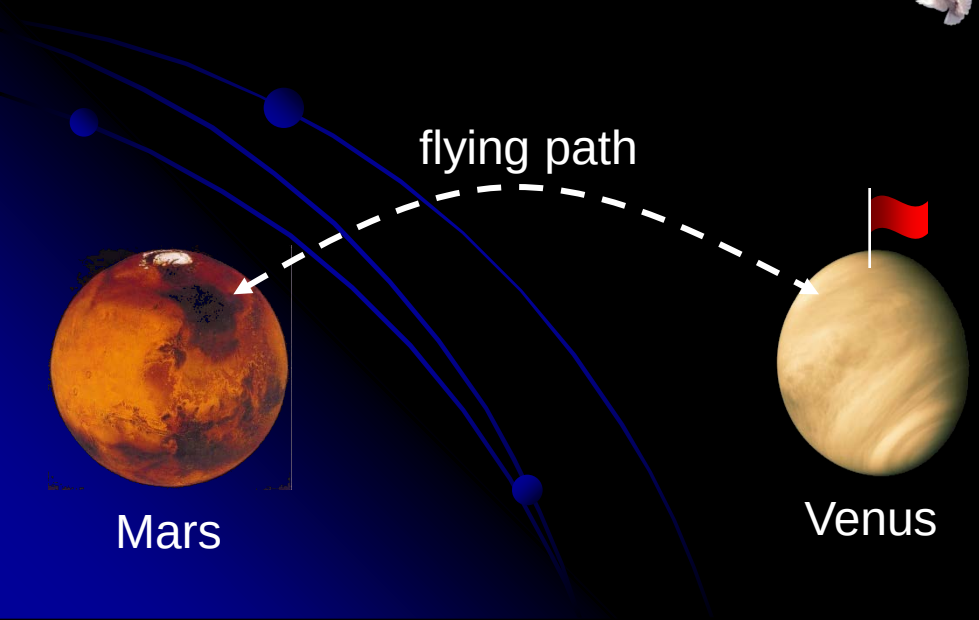
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

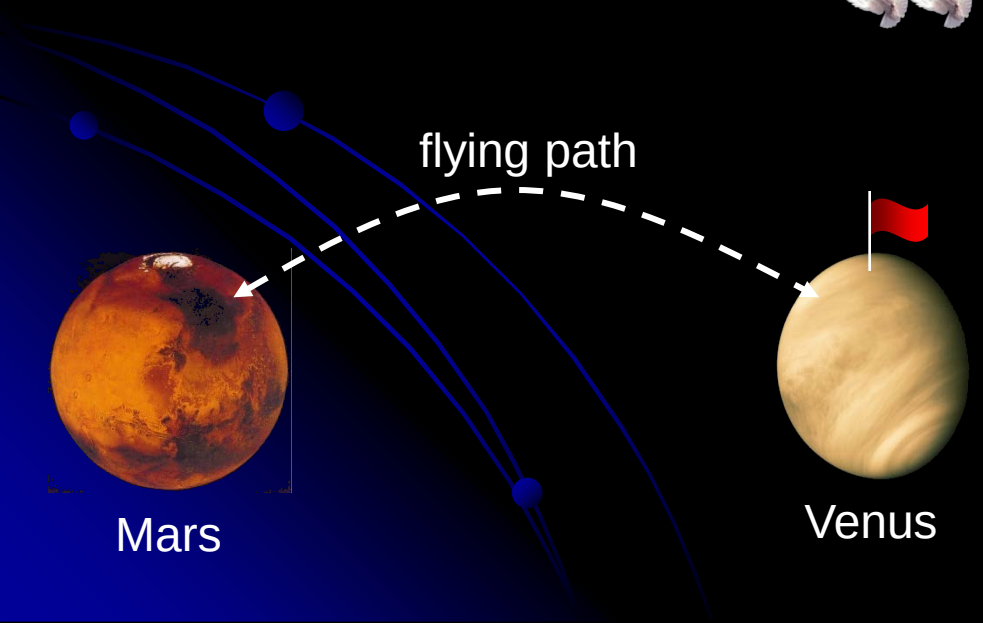
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

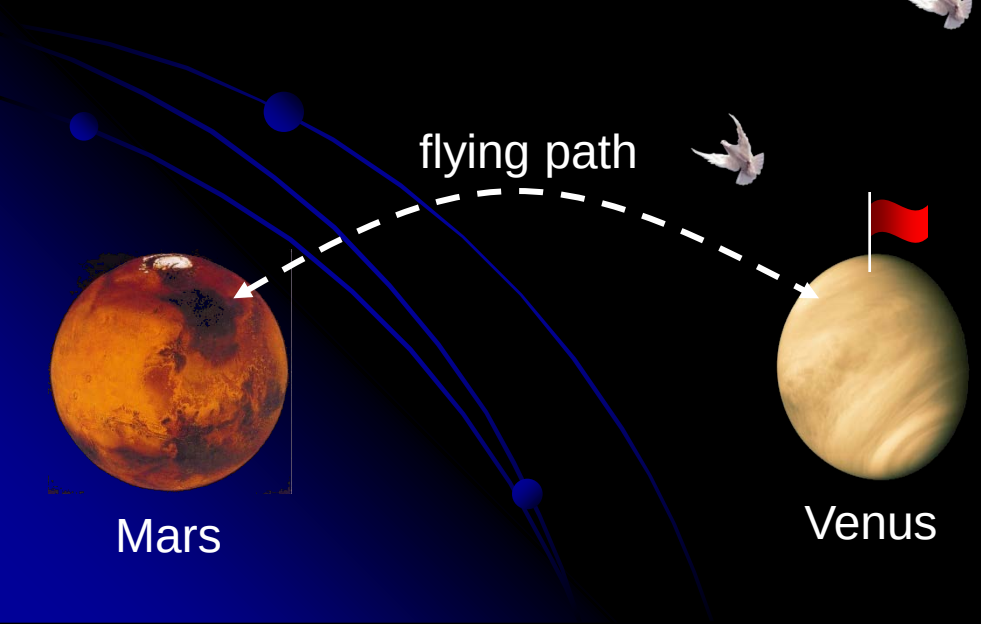
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

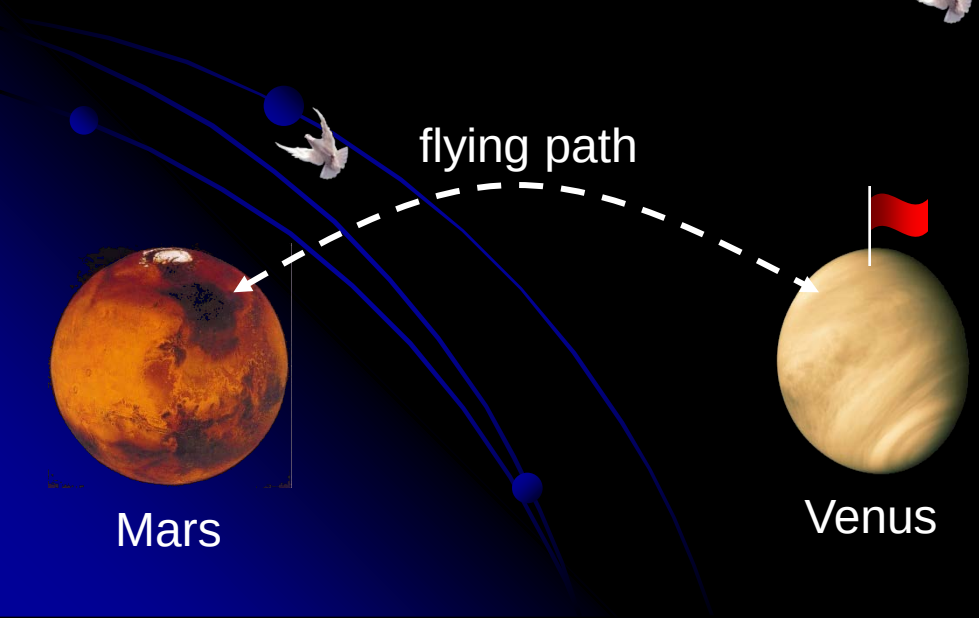
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

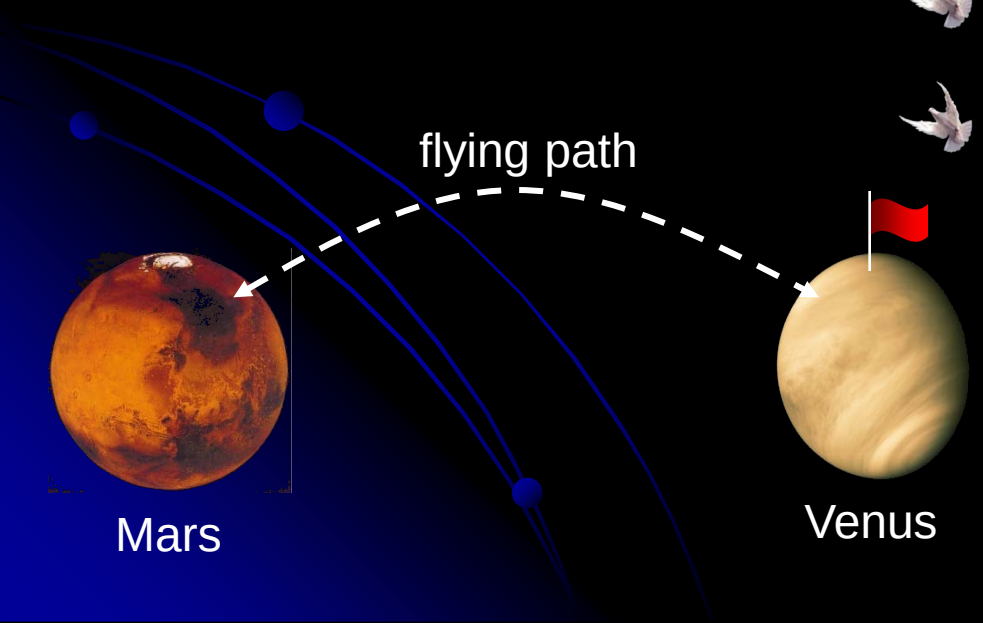
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

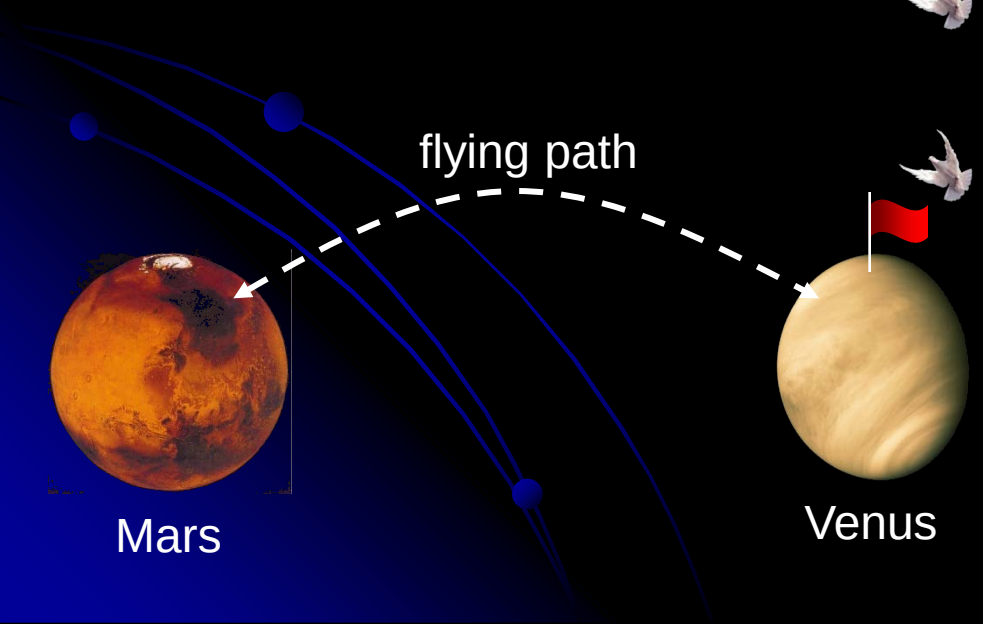
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

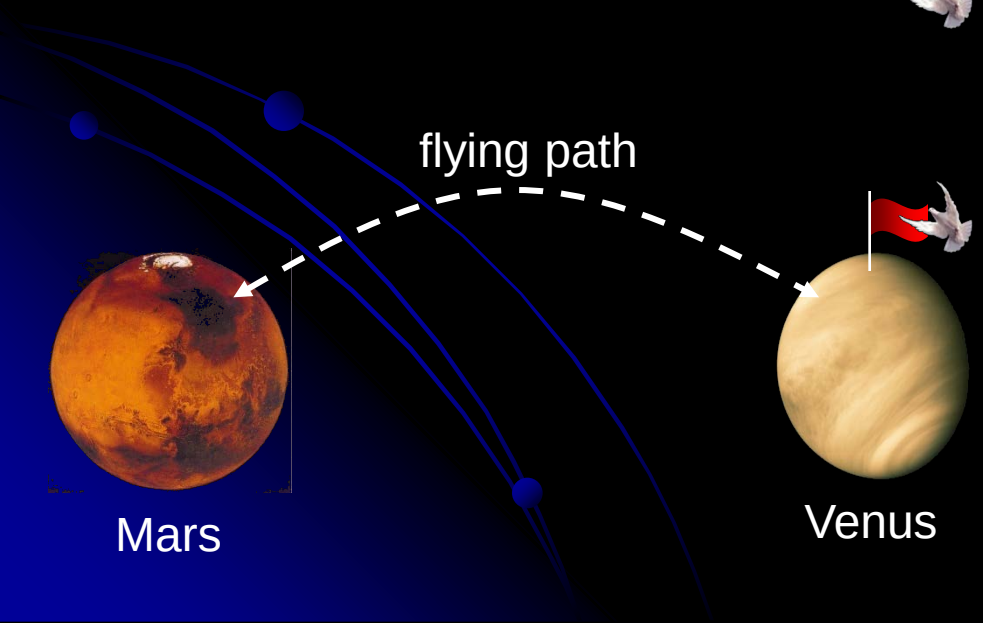
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

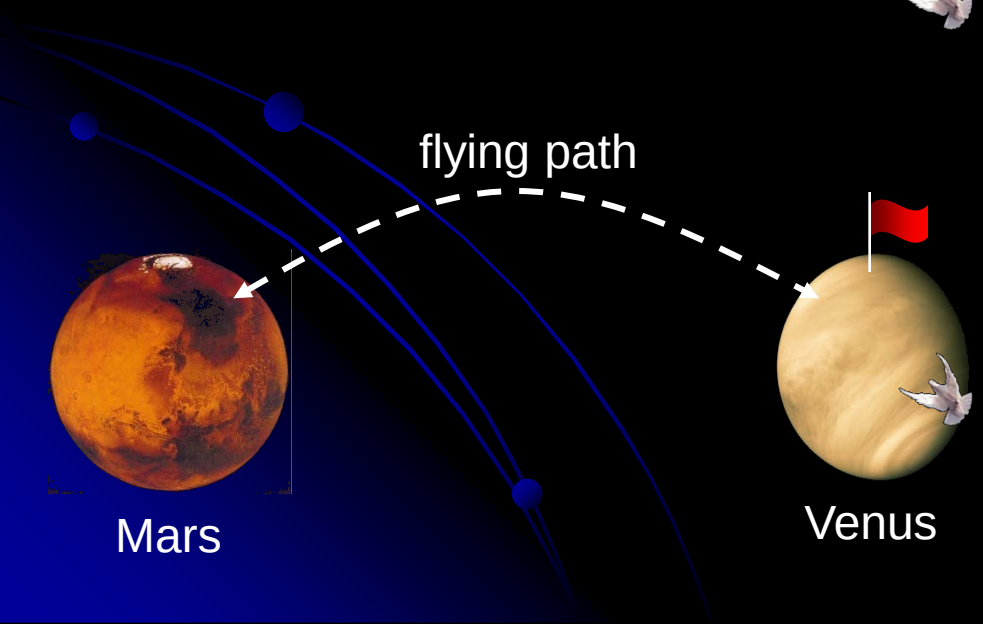
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

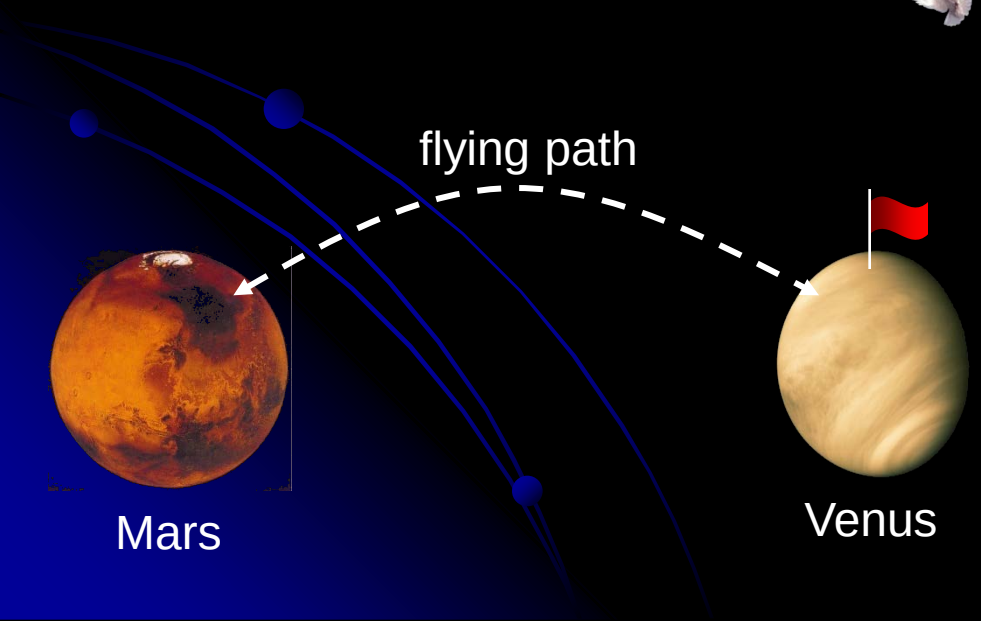
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

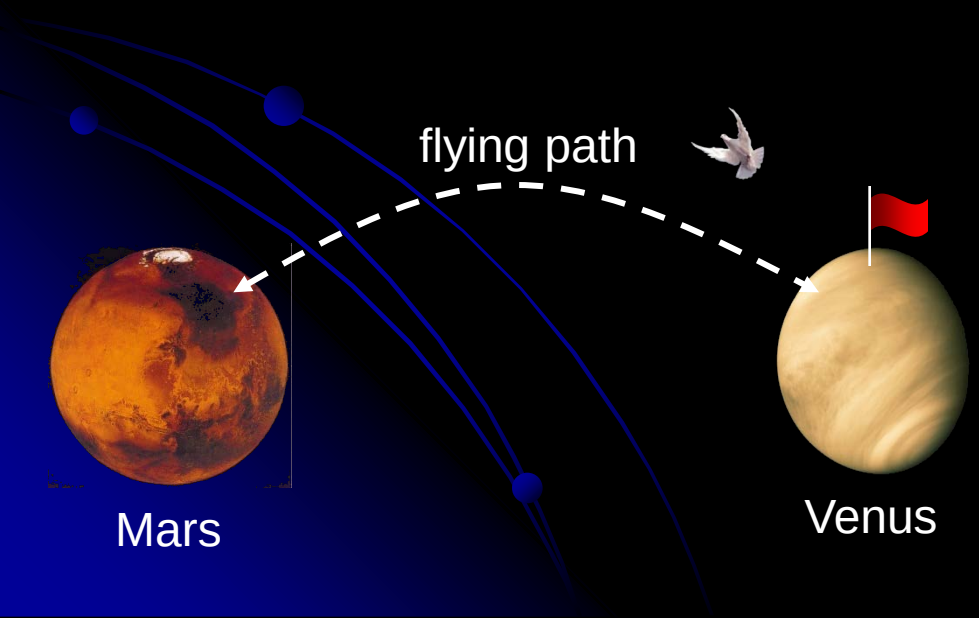
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

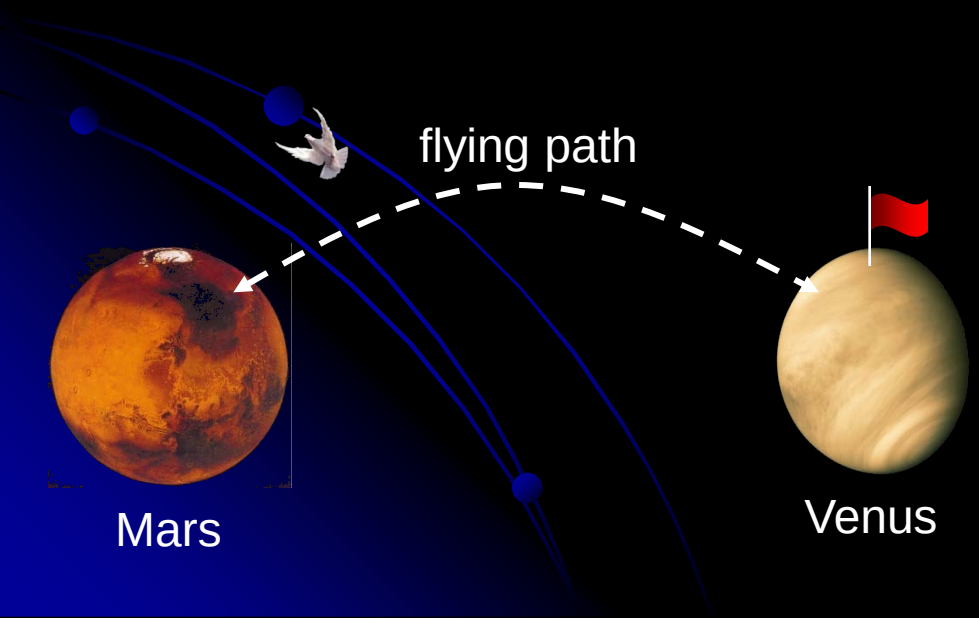
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

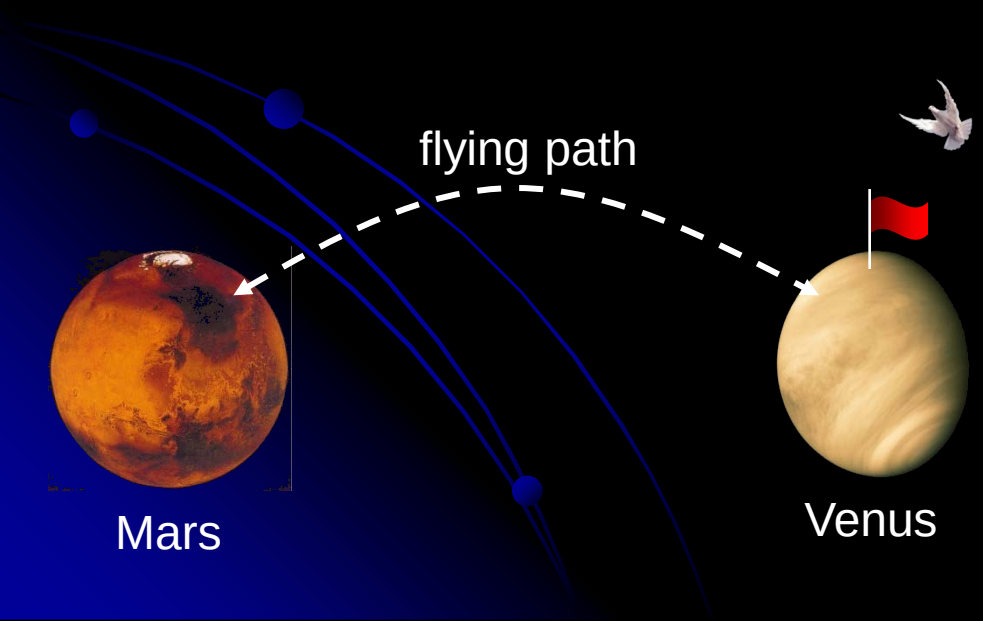
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

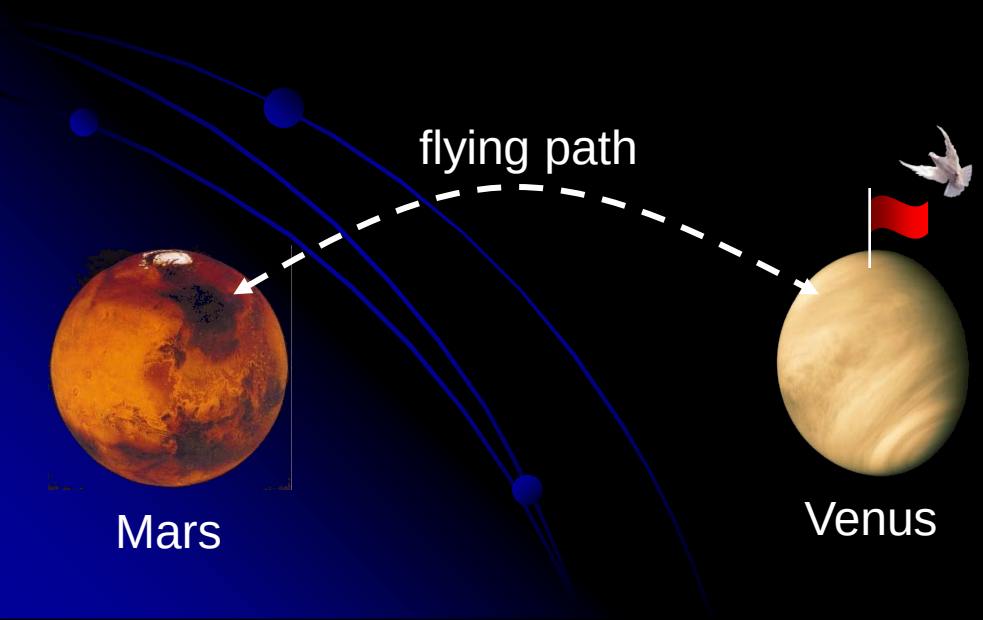
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

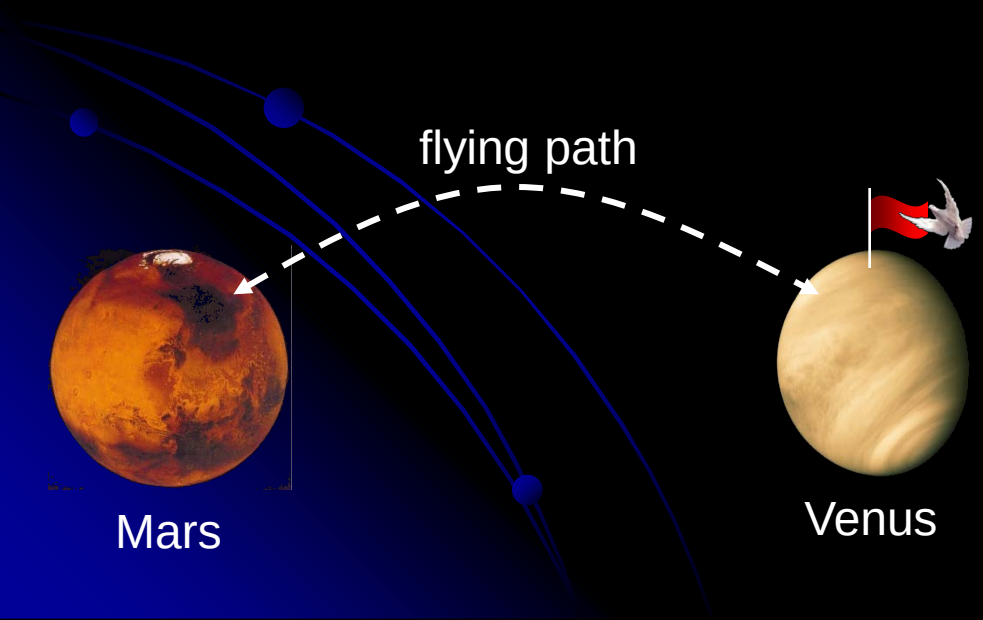
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

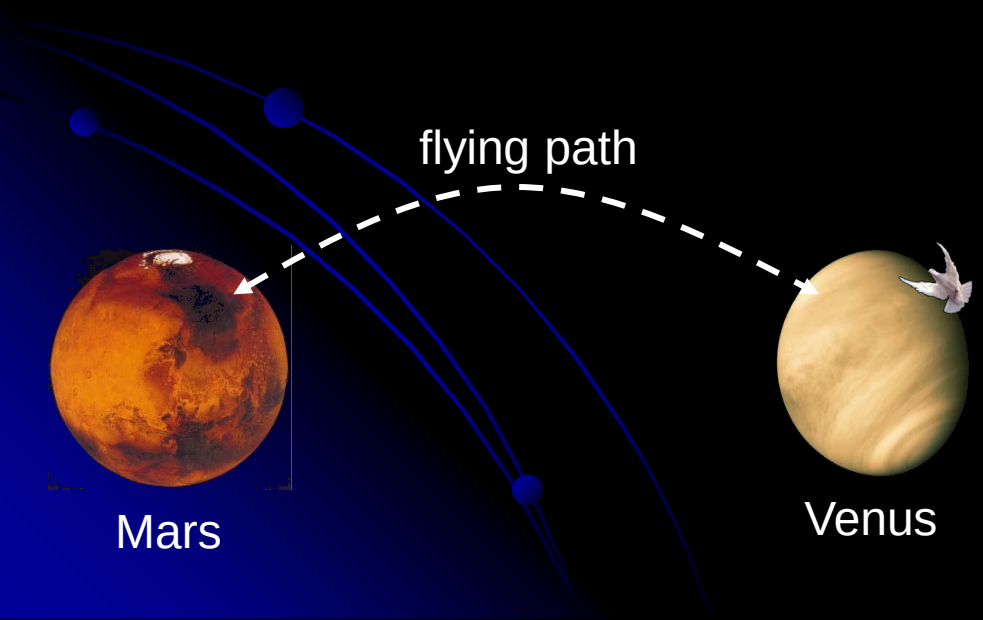
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

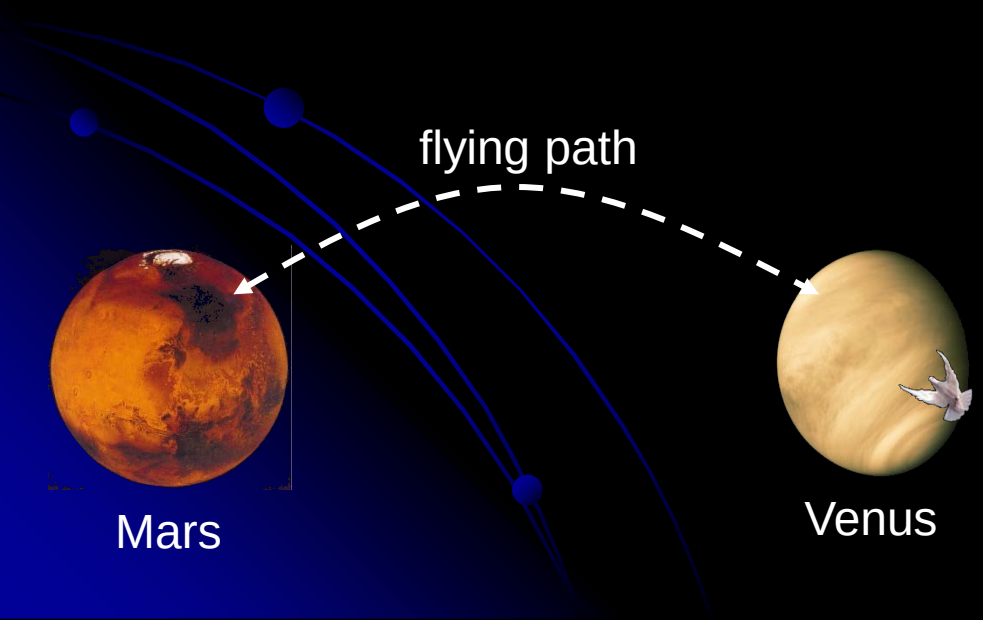
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

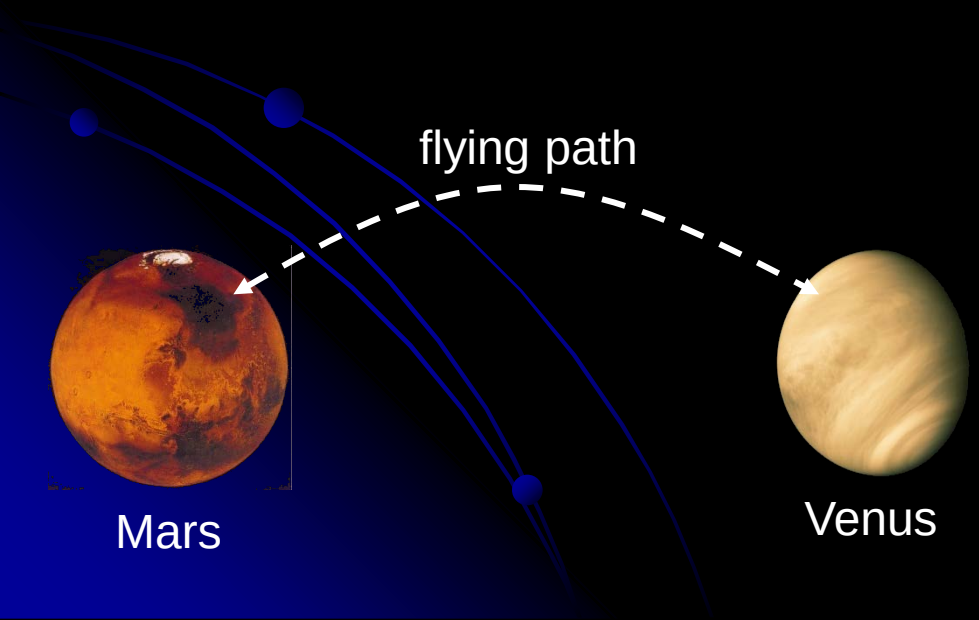
```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

```
semaphore flyingPath = 1;

P(flyingPath);
// send the message
V(flyingPath);
```

```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```



Allowing Multiple Venusians in Transit Revised Again

- Efficient for Venusians
- Not fair for Martians

Allowing Multiple Venusians and Martians in Transit

```
semaphore flyingPath = 1;
semaphore MartianLock = 1;
int MartiansInTransit = 0;

P(MartianLock);
++MartiansInTransit;
if (MartiansInTransit == 1) {
   P(flyingPath);
}
V(MartianLock);
// send the message
P(MartianLock);
--MartiansInTransit;
if (MartiansInTransit == 0) {
   V(flyingPath);
}
V(MartianLock);
```

```
semaphore VenusianLock = 1;
int VenusiansInTransit = 0;

P(VenusianLock);
++VenusiansInTransit;
if (VenusiansInTransit == 1) {
   P(flyingPath);
}
V(VenusianLock);
// send the message
P(VenusianLock);
--VenusiansInTransit;
if (VenusiansInTransit == 0) {
   V(flyingPath);
}
V(VenusianLock);
```

Allowing Multiple Venusians and Martians in Transit

- Both sides can transmit efficiently
- One side can completely block off the traffic from the other side
- Fair, since both sides are equally selfish…
- Fortunately, pigeons do sleep
- Daily cycles will break a starvation condition…

Proving the Correctness

- ***Safety (mutual exclusion)***:
  - At most one thread is in the critical section at any time
  - All necessary locks are acquired at the entrance of a critical section
  - All shared resources are protected

Proving the Correctness

- ***Liveness (progress)***:
  - If more than one thread is interested in executing the critical section, some processes will eventually do so (deadlock free)
- ***Fairness (bounded waiting)***:
  - Every thread that wants to execute the critical section will eventually do so (no starvation)