

Futures und Promises

Umsetzung in verschiedenen Programmiersprachen

Futures und Promises

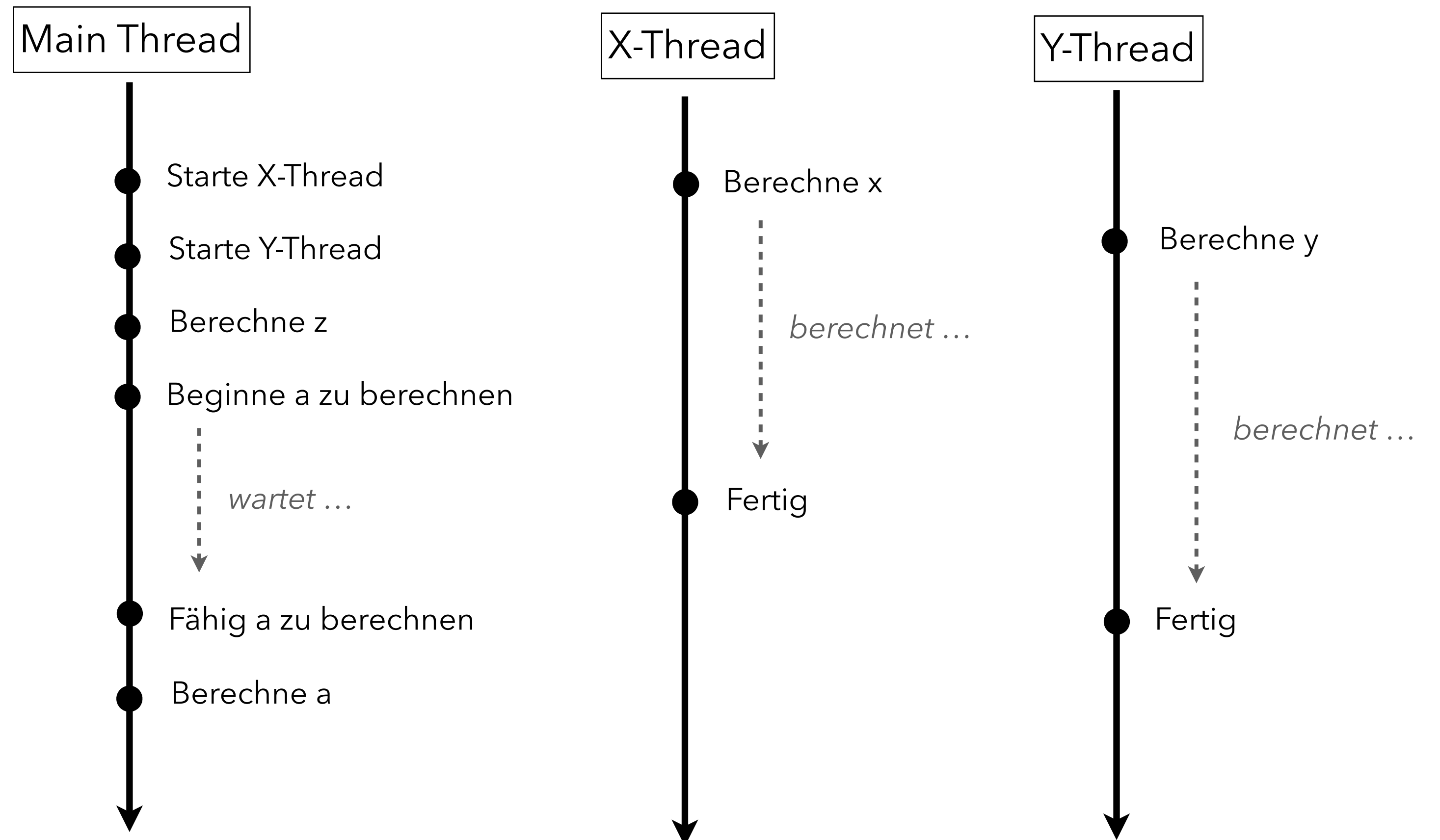
Definition

- *High Level Concurrency Konstrukt* zur Unterstützung asynchroner Programmierung
- Das Future ist ein Platzhalter für eine Berechnung, dessen Ergebnis noch nicht existiert
 - Wert wird parallel bereitgestellt um anschließend verwendet zu werden.
- Promise ist ein Versprechen das jenes Ergebnis eintreten wird

Futures und Promises

Beispiel

```
x @= calc_x()  
y @= calc_y()  
z := calc_z()  
  
a := x + y + z
```



Futures und Promises

Go

```
func future(function func() (int, bool)) Future {  
    channel := make(chan futureType)  
    go func() {  
        value, status := function()  
        fType := futureType{value, status}  
        channel <- fType  
    }()  
    return channel  
}
```

Futures und Promises

Go



```
func div(a int, b int) Future {  
    return future(func() (int, bool) {  
        if b > 0 {  
            return (a / b), true  
        } else {  
            return 0, false  
        }  
    })  
}
```

Futures und Promises

C++

```
void add(std::promise<int>&& promise, int a, int b) {  
    promise.set_value(a+b);  
}
```

```
std::promise<int> promise;  
std::future<int> future = promise.get_future();  
std::thread thread(add, std::move(promise), a, b);  
int future = future.get();  
thread.join();
```

Dankeschön

Fragen

Raphaelle Licciardo - 60559