Alexander Jenke, Theodor Straube

Übung12: Multithreading

Programmierkurs Python // Mittwoch, 29. Januar 2020

TOC

Prozesse / Threads

- (Unix) **Prozess**:

  „program in execution“

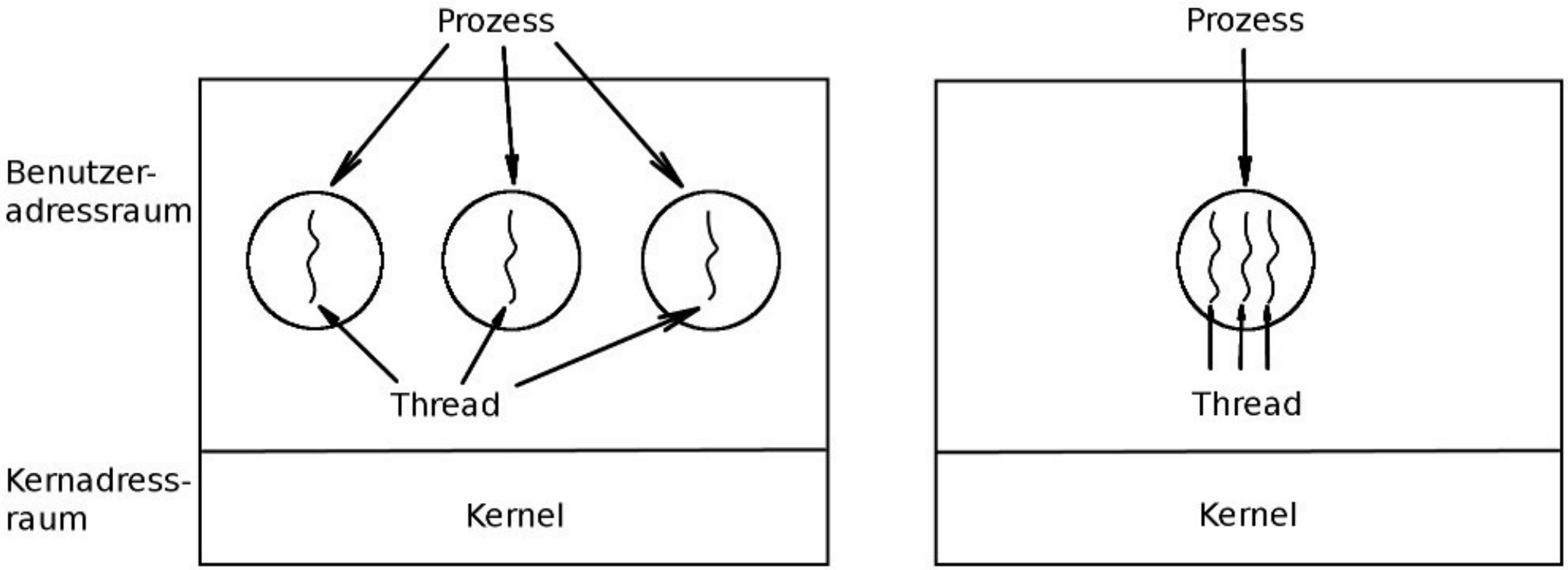
→ vereint laufendes Programm, einen Adressraum und Betriebsmittel (z.B. Dateien)[1]

- **Thread**:

  „Selbständige, ein sequentielles Programm ausführende, zu anderen Threads parallel arbeitende, vom Betriebssystem zur Verfügung gestellte Aktivität."[1]

Bild: Didia, https://commons.wikimedia.org/wiki/File:Process-thread.jpg

Prozesse / Threads



Threads in Python
threading

- Modul in der Python Standard Library
- Bietet die Klasse threading.Thread, mit den Funktionen:
  - `start`: Startet den Thread
  - `is_alive`: Gibt True zurück, wenn die Ausführung begonnen hat aber noch nicht abgeschlossen ist
  - `join`: Wartet, bis der Thread terminiert
- Instanziierung mit dem `target`-Keyword oder Vererbung

```python
import threading


def foo(parameter=10):
    print(parameter)


new_thread = threading.Thread(target=foo, kwargs={'parameter': 10})
```

```python
class FooThread(threading.Thread):

    def __init__(self, parameter=10):
        super().__init__()
        self.parameter = parameter

    def run(self):
        foo(parameter=self.parameter)


new_thread = FooThread(parameter=20)
```

Threads in Python
concurrent.futures

- Interface für parallele Ausführung
- Teil der Python Standard Library, verfügbar ab Version 3.2
- Enthält die `ThreadPoolExecutor`-Klasse, mit der sich Threads gesammelt verwalten lassen:
  - Verfügbar über einen Context Manager
  - Mit submit können einzelne Threads in Auftrag gegeben werden

    Diese Aufrufe geben ein `Future`-Objekt zurück, das Informationen über den Zustand des Threads gibt

```python
import concurrent.futures

results = []
with concurrent.futures.ThreadPoolExecutor() as executor:

    futures = [executor.submit(foo, parameter) for parameter in range(100)]

    for future in concurrent.futures.as_completed(futures):
        results.append(future.result())
```

Threads - Probleme

- Gemeinsame Resourcen nutzen → Race Conditions
- Python wird nicht parallel ausgeführt
- Parallel ist nicht dasselbe wie nebenläufig
- Multiprocessing

nächste Woche

- Mehr Zeit für Aufgaben
- Zeit für Fragen
- Lösungen