

# Systemprogrammierung

## **System Dokumentation**

Dozent: Prof. Dr. Ammar Memari

vorgelegt von: Anael Ngome 6020606  
Frank Williams Nganmo Tchipsse 605321  
Nidaa  
Cafet Sentürk 6054150  
Sara

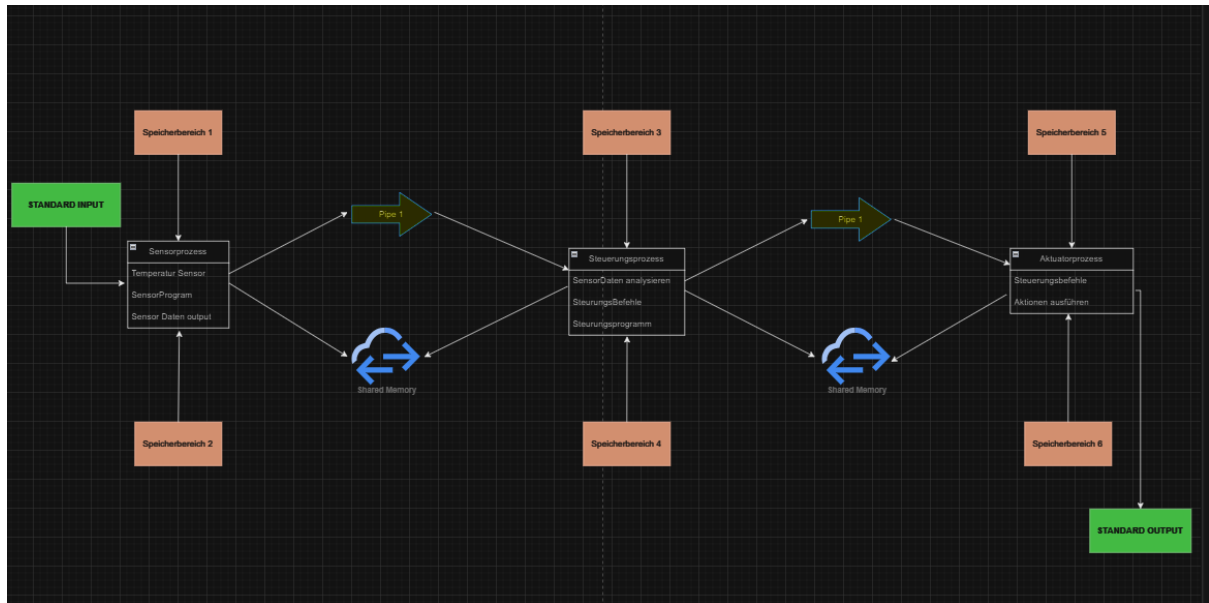
Abgabetermin: 18-12-2023

## Table of Contents

<b>1. Systementwurf.....</b>	<b>3</b>
1.1 Systemarchitektur:.....	3
1.2 Definition der Datenstrukturen für die Interprozesskommunikation: .....	3
<b>2. Implementierungsdetails.....</b>	<b>4</b>
<b>3. Nutzung der Interprozesskommunikation .....</b>	<b>4</b>

# 1. Systementwurf

## 1.1 Systemarchitektur:



## 1.2 Definition der Datenstrukturen für die Interprozesskommunikation:

### Die zu übertragende Informationen:

- Messwerte vom Sensor
- Steuerbefehle von der Steuerung und
- Aktuatorstatusinformationen

### Deren Datenstruktur:

- Messwerte vom Sensor: Ganzzahlen
- Steuerbefehle von der Steuerung: Ganzzahlen
- Aktuatorstatusinformationen: String

(Ganzzahlen, Gleitkommazahlen oder Zeichenketten sein, oder komplexere Datenstrukturen wie Arrays, Strukturen oder sogar benutzerdefinierte Klassen/Objekte.)

### Berücksichtigung der IPC-Methode

- Empfangenen Daten aus der Shared Memory: float

- Empfangene Daten aus den Pipes: float

## 2. Implementierungsdetails

### Definition der Protokolle und Schnittstellen

- wie die verschiedenen Prozesse miteinander kommunizieren werden.
- welche Prozesse welche Daten senden und empfangen werden.
- Definieren die Schnittstellenklar.

## 3. Nutzung der Interprozesskommunikation

### a. Pipes:

- **Sensorprozess (Temperaturprozess):**
  - Verwendet eine Pipe (t.pipe\_fd) für die Kommunikation mit dem Steuerungsprozess.
  - Schreibt die simulierten Temperaturdaten in die Pipe (write(t.pipe\_fd[1], &t.temperature, sizeof(float))).
- **Steuerungsprozess:**
  - Erzeugt eine Pipe (sensor\_pipe) für die Kommunikation mit dem Sensorprozess und AktuatorProzess.
  - Liest die Temperaturdaten aus der Pipe (read(sensor\_pipe[0], &temperature, sizeof(float))).
  - Sendet Befehle an den Aktuatorprozess durch Schreiben in die Pipe (write(pipe\_actuator[1], &schaltung\_control, sizeof(int))).

### b. Shared Memory:

- **Sensorprozess:**
  - Erzeugt einen Shared Memory-Bereich (t.shm\_id) für die Kommunikation mit dem Steuerungsprozess.
  - Liest Befehle aus dem Shared Memory (a.received\_commands = \*a.shared\_data).
- **Steuerungsprozess:**
  - Erzeugt einen Shared Memory-Bereich (shm\_id) für die Kommunikation mit dem Aktuatorprozess.

- Schreibt Befehle in den Shared Memory (\*shm\_data = control\_command).
- liest Befehle in den Shared Memory (\*t.shm\_data = control\_command).

Insgesamt erfolgt die Kommunikation wie folgt:

1. Der Sensorprozess generiert Temperaturdaten und schreibt sie in die Pipe und liest Shared Memory.
2. Der Steuerungsprozess liest die Temperaturdaten aus der Pipe und schreibt Shared Memory, bewertet die Temperatur und schreibt Befehle in die Pipe und den Shared Memory.
3. Der Aktuatorprozess liest die Befehle aus der Pipe und dem Shared Memory und führt entsprechende Aktionen aus.