

Übung zu Betriebssystemtechnik

Aufgabe 6: Nachrichtenaustausch

02. Juli 2025

Dustin Nguyen, Maximilian Ott & Phillip Raffeck

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



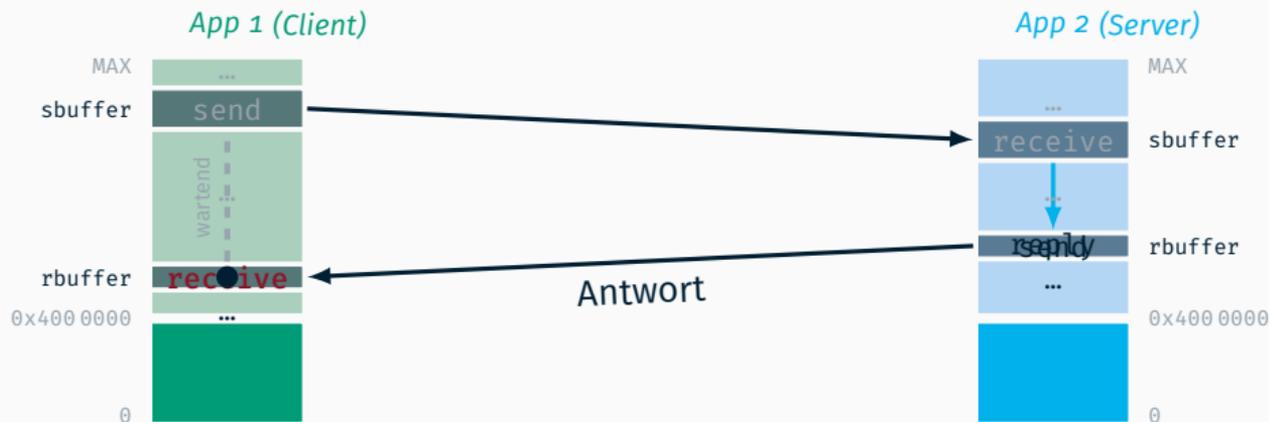
Lehrstuhl für Informatik 4
Systemsoftware



Friedrich-Alexander-Universität
Technische Fakultät

Prozesse sollen über Systemaufrufe miteinander kommunizieren (Nachrichten austauschen) können

Interprozesskommunikation



send

- versendet eine Nachricht an einen Empfangsprozess
- erwartet den Nachrichteneingang

receive

- empfängt eine Nachricht von einem Sendeprozess
- erwartet den Nachrichteneingang

Implementierung

Threads müssen für die Interprozesskommunikation

- eindeutig identifizierbar sein

`pid` aus Prozessverwaltung (von *Aufgabe 5*)

- Nachrichten verwalten (Queues)

`msg_inbox` (unbearbeitete) Aufträge

`msg_reply` ausstehende Antworten (zu empfangenen Aufträgen)

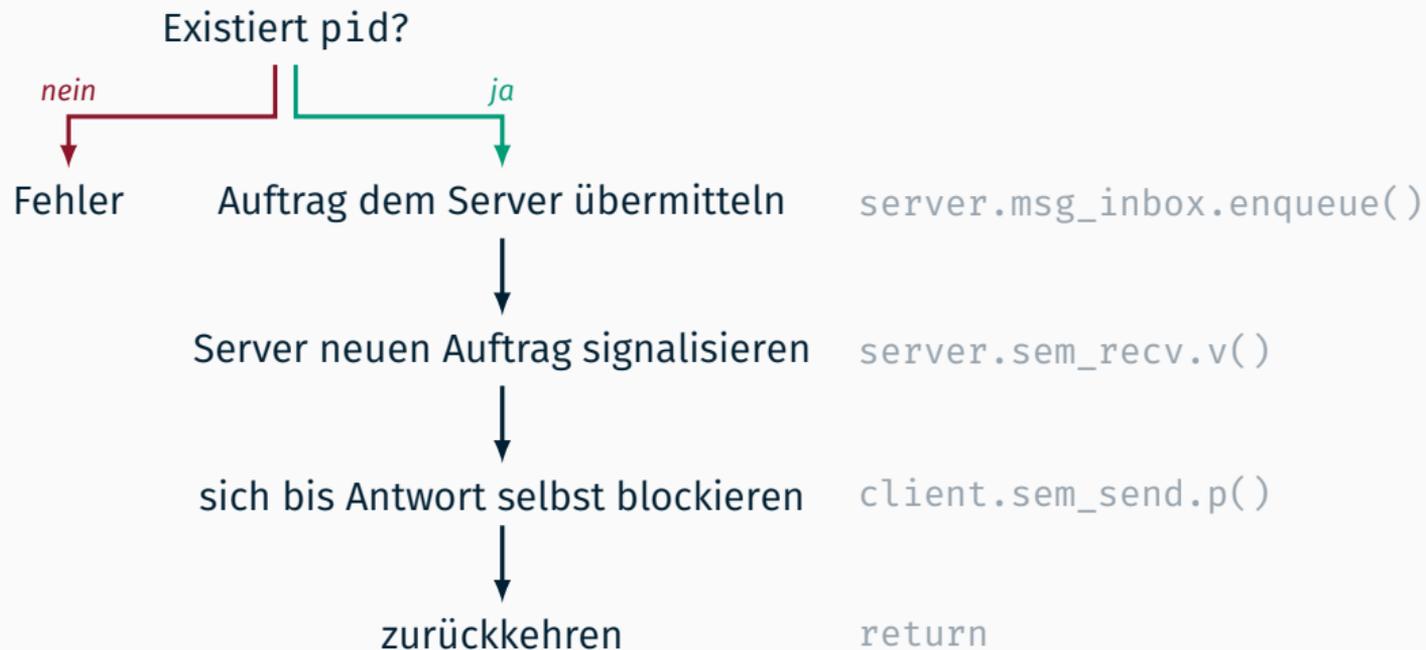
- blockieren und signalisieren

`sem_recv` Empfang einer neuen Nachricht (Auftrag)

`sem_send` abgeschlossene Bearbeitung einer Nachricht
(Antwort vorhanden)

Ablauf Sendeoperation

```
void send(int pid, const void *sbuffer, size_t ssize,  
          void *rbuffer, size_t rsize);
```



Ablauf Empfangsoperation

```
int recv(void *sbuffer, size_t ssize);
```



Ablauf Antwortoperation

```
void reply(int pid, const void *rbuffer, size_t rsize);
```

Gab es einen Auftrag von pid?

```
server.msg_reply.contains()
```

nein

ja

Fehler

(Server-)Puffer zu Client kopieren

```
rbuffer.copy()
```

Client die Antwort signalisieren

```
client.sem_send.v()
```

zurückkehren

```
return
```

Testen von Aufgabe (5 &) 6

```
char sbuf[8194], rbuf[8194];
void main() {
    fork();
    fork();
    if (fork() == 0) {
        sbuf[0] = 3;
        char d = sbuf[8192] = getpid() % 22;
        sbuf[8193] = 1;
        send(getppid(), sbuf, 8193, rbuf, 8193);
        char m[] = "Reply A=A bad!\ngood";
        m[6] += rbuf[0] + sbuf[8193];
        m[8] += 4 + d;
        for(size_t i = 0; m[6] == m[8] && i < 4; i++) m[i + 10] = m[i + 15];
        write(0, m, 15);
    } else {
        int X = recv(rbuf, 8193);
        rbuf[0] = rbuf[0] + rbuf[8192];
        rbuf[8193] = 7;
        reply(X, rbuf, 8193);
    }
    exit();
}
```

Fragen?

Nächste Woche folgt die Tafelübung zur letzten Aufgabe