



## Sys. Inf. III (Betriebssysteme)

Wintersemester 2004/05

Serie 9

13. Dezember 2004

**Thema: Message passing, Speicherverwaltung**

**Ausgabetermin: 13. Dezember 2004**

**Abgabe: 20. Dezember 2004**

**Aufgabe 1 (Rendez-Vous (8 Punkte))** Wir haben in der Vorlesung *message passing* als eine Möglichkeit der Prozeßkommunikation behandelt (Kapitel 7 aus [Com83]). Die Implementierung in Xinu ist nur eine (recht einfache) Variante wie sich Kommunikation durch Nachrichtenaustausch in Betriebssystemen realisieren läßt. Eine Variante des *message passing* ist Kommunikation durch *Rendez-Vous*-Kommunikation. Ihr Kennzeichen ist, daß sowohl das Senden wie das Empfangen *blockierend* sind.

Nehmen wir an, wir hätten —wie beim *message-passing* Konzept von Xinu— einen Sendeprozess und einen Empfangsprozess, dann “blockiert” der Sender einer Nachricht solange, bis der Empfänger sie liest und umgekehrt.

Implementieren sie zwei Systemaufrufe

- SYSCALL `sendrv(pid, msg)` und
- SYSCALL `receiverv(pid)`

mit dem beschriebenen Verhalten. Beachten Sie, daß der Systemaufruf für das Empfangen nun den Sender als Parameter bekommt. Überlegen Sie, warum das im Zusammenhang mit *Rendez-Vous*-Kommunikation sinnvoll ist. Überlegen Sie auch, was passieren soll, wenn ein Prozess sich selbst eine *Rendez-Vous*-Nachricht zukommen lassen will.

**Aufgabe 2 (Speicherverwaltung (2 Punkte))** Xinu meldet einen Fehler, falls ein Prozess mehr Speicher anfordert als gegenwärtig vorhanden. Ähnlich wie bei Semaphoren könnte man versuchen, in diesem Fall den Prozess, der den Speicher braucht, solange zu verzögern, beispielsweise in dem man ihn in eine weitere Warteschlange einreicht, bis genügend Speicher freigegeben worden ist. Erklären Sie, wie es unter diesen Umständen passieren kann, daß alle Prozesse auf Speicher warten, obwohl genügend vorhanden ist.

## Literatur

[Com83] Douglas Comer. *Operating System Design, The Xinu Approach*. Prentice Hall, 1983.