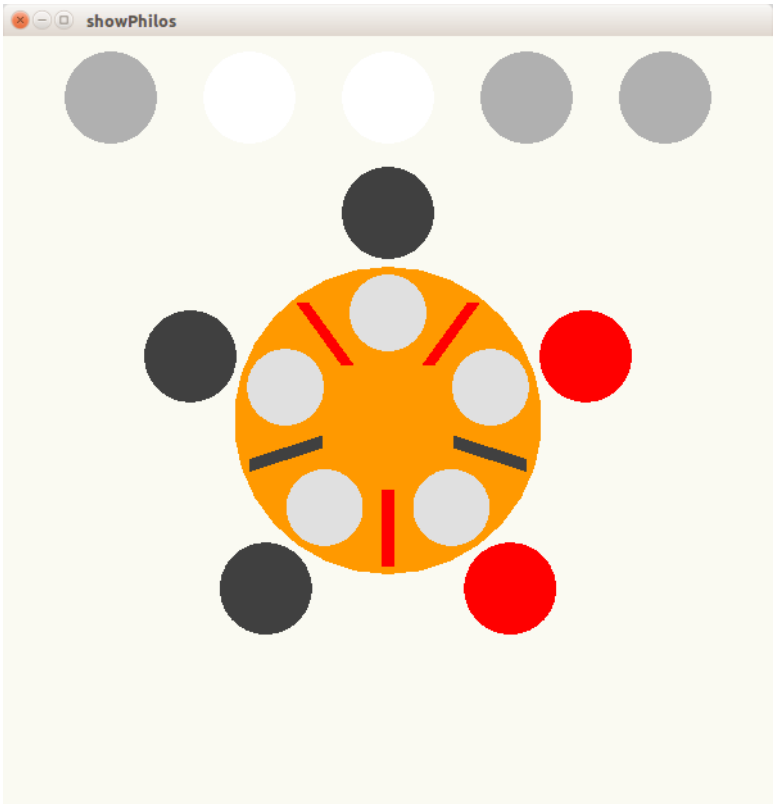
Unix Inter-Process-Communication II (POSIX:SEM und POSIX:XSI IPC)

Das sogenannte „Dining Philosophers Problem“ (http://de.wikipedia.org/wiki/Philosophenproblem) soll mit Prozessen gelöst werden.

Fünf Philosophen wechseln ihren Zustand zwischen *denkend* und *essend*. Zum Essen hat jeder an einem runden Tisch einen Platz und einen Teller, zwischen jedem Teller liegt eine Gabel (oder ein Esssstäbchen, *chopstick*). Allerdings werden zum Essen zwei Gabeln benötigt - die jeweils links bzw. rechts neben dem Teller. So können höchstens zwei Philosophen gleichzeitig essen.

Im Ordner `UE08/Vorgabe` finden Sie ein Grafikprogramm `showPhilos.c`, das die Situation darstellt. Eventuell müssen Sie zunächst OpenGL installieren (s. Kommentar in der Datei).

Wenn Sie das ebenfalls gegebene Programm `showPhilosTest.c` gleichzeitig starten, sehen Sie wie die Philosophen zwischen dem *Denkraum* (oben quer, hellgrau) und den Stühlen am Esstisch wechseln (sitzende Philosophen werden rot). Die Gabeln wechseln ihren Zustand zwischen schwarz(*frei*) und rot(*belegt*). Hier eine **ungültige** Situation:



Die beiden Prozesse kommunizieren über gemeinsamen Speicher (POSIX:XSI shared memory), wobei der Zugriff auf diesen kritischen Bereich mit einem benannten Semaphor (POSIX:SEM named semaphor) geschützt wird.

Zur Lösung des einst von E.W.Dijkstra formulierten Philosophenproblems existieren diverse Vorschläge, wobei die Korrektheit aber nicht immer gesichert ist.
Die naheliegende Lösung wäre: jede Gabel wird mit einem Mutex/Semaphor exklusiv geschützt (Stühle müssen nicht geschützt werden, jeder Philosoph hat seinen festen Platz). Möchte ein Philosoph essen, so belegt er z.B. erst seine linke, dann seine rechte Gabel (*Linksstarter*). Wechselt er in den Denkraum,

gibt er die Gabeln wieder frei. Diese Lösung führt zu einem deadlock, wenn die Philosophen reihum ihre erste Gabel belegt haben.

Macht man die Philosophen abwechselnd zu *Rechts*- und *Linksstartern*, ist das Problem gelöst. Mit diversen anderen Lösungen finden Sie diesen Vorschlag als C-Lösung z.B. hier:
http://rosettacode.org/wiki/Dining_philosophers

Aufgabe 1: philos

Erstellen Sie ein Programm `philos.c`, welches das Philosophenproblem korrekt löst und die Zustandsänderungen dem obigen Grafikprogramm über den gemeinsamen Speicher so mitteilt, dass eine Animation der Lösung dargestellt wird. Folgende Implementation ist hier vorgesehen:

- Das Grafikprogramm soll nicht verändert werden. Es liest den gemeinsamen Speicher alle 100 Millisekunden aus und zeichnet ein neues Bild.
- Das Programm `philos.c` soll fünf child-Prozesse erzeugen - einen pro Philosoph.
- Jeder Philosoph legt selbst die Dauer seines Tisch- bzw. Denkraumaufenthalts per Zufall fest (z.B. 2-3 Sekunden, gegebene Funktion `schlafzufaellig()` verwenden).
- Der gemeinsame Speicher soll als kritischer Bereich geschützt werden. Hierzu muss der im Grafikprogramm bereits verwendete Semaphor `/fourtytwo` genutzt werden.
- Die *Gabel*-Semaphoren können mit benannten POSIX:SEM Semaphoren oder einer POSIX:XSI Semaphorgruppe umgesetzt werden.

ACHTUNG: Diese Semaphoren sind persistent und behalten ihren Zählwert nach Prozessende. Beim erneuten Start des Prozesses wird dann evtl. auf einen gesperrten Semaphor gewartet. Die Semaphoren sollten daher am Prozessende entfernt werden. Das gegebene Grafikprogramm löscht den dort verwendeten Semaphor im SIGINT-handler und im keyboard-handler für ESC. Daher immer mit ESC oder Strg-C beenden.

-> Auch Ihr Programm sollte die *Gabel*-Semaphoren unbedingt bei Prozessende entfernen.

Aufgabe 2: philos2

Folgende Änderungen sollen vorgenommen werden:

- Die Philosophen bestimmen die Denk- und Esszeiten nicht mehr selbst, vielmehr bestimmt der main-Prozess diese Zeiten für alle fünf Philosophen.
- Der main-Prozess erzeugt die Zufallszeiten und sendet sie mit einer POSIX:XSI message queue gezielt an die einzelnen Philosophen.
- Für das „gezielte“ Versenden soll der Nachrichtentyp der queue verwendet werden.

Hinweis: Es müssen nur die Essenszeiten übermittelt und modelliert werden. Die Denkzeiten ergeben sich als Wartezeiten an der leeren queue.