

OPERATING SYSTEMS BEISPIEL 3

Aufgabenstellung – 4wins

Wieder einmal wissen Sie nichts mit Ihrer Zeit anzufangen, wollen aber auch nicht aus purer Langeweile eine Partie „4 Gewinnt“ gegen sich selbst verlieren. So implementieren Sie es, kurz entschlossen, einfach in C; und zwar so, dass auch Sie eine Chance haben zu gewinnen.

Anleitung

SYNOPSIS:

```
4wserver
4wclient
```

Schreiben Sie zwei Programme (*Server* und *Client*), die miteinander über ein Shared Memory kommunizieren und sich mit Hilfe von Semaphoren synchronisieren. Das Spiel besteht aus einem Server und nur einem Client, der Server muss also nicht mehrere Clients gleichzeitig unterstützen. Das (virtuelle) Spielfeld soll eine Größe von 7×6 haben, und nur von oben „befüllbar“ sein (genau wie das Originalspiel). Der Server erzeugt das Shared Memory und verwaltet eine interne Kopie vom enthaltenen Spielbrett, damit ein Schummeln (mehrere Einwurfe oder unerlaubte Spielfeldveränderungen) erkennbar ist. Außerdem soll der Server, als Computergegner, nach jedem gültigen Einwurf des Clients selbst einwerfen; dabei sollen folgende Regeln berücksichtigt werden:

- Mit einer Wahrscheinlichkeit von $1/3$ soll dort eingeworfen werden, wo der menschliche Gegner zuletzt eingeworfen hat.
- Mit einer Wahrscheinlichkeit von $2/3$ soll zufällig eingeworfen werden.
- Ist die ausgewählte Spalte bereits voll, soll eine gefunden werden, die noch Platz hat – gibt es keine mehr, und keinen Gewinner, steht es unentschieden.

Wird ein Sieg, Unentschieden oder Schummeln erkannt, gibt das der Server dem Client bekannt; der Client zeigt bei einem Sieg oder Unentschieden nochmals das Spielbrett an und startet das Spiel dann, nach Betätigung der Entertaste, neu. Beim Schummeln soll das Spiel gleich neu gestartet werden, ohne dass das Spielfeld nochmals angezeigt wird.

Der Client soll das Spielfeld anzeigen und über *stdin* in einer Endlosschleife Kommandos entgegen nehmen. Nach jedem Kommando, das sich auf das Spielfeld auswirkt, soll das Spielfeld erneut angezeigt werden. Arbeiten Sie mit *system(3)* und dem Shell-Befehl *clear*, um den Bildschirm zu löschen.

Kommandos:

```
d<spalte nr>   wirf bei <spalte nr> ein, z.B. d2 oder d5
u<spalte nr>   nimm Zug bei <spalte nr> zurueck, z.B. u2 oder u5
c              beende Einwurf und gib das Spiel an den Computergegner zurueck
n              Neustart des Spiels
q              Beenden
```

Wird ein anderes (unbekanntes) Kommando eingegeben, soll eine Auflistung der gültigen Kommandos ausgegeben werden.

Beim Beenden des Clients soll das Spiel erhalten bleiben, sodass es beim erneuten Start des Clients weitergespielt werden kann.

Hinweise

Um eine Kommunikation zwischen Server und Client abseits des Spielfelds zu ermöglichen (Spielstatus o.Ä.), ist es ratsam, außer dem Array noch eine Variable im Shared Memory zu definieren, um Statusnachrichten auszutauschen. Dafür sind **structs** ganz nützlich.

Richtlinien

Bitte beachten Sie auch die *Richtlinien für die Erstellung von C-Programmen* sowie die *Allgemeinen Hinweise zur Beispielgruppe 3* auf der Übungswebsite.