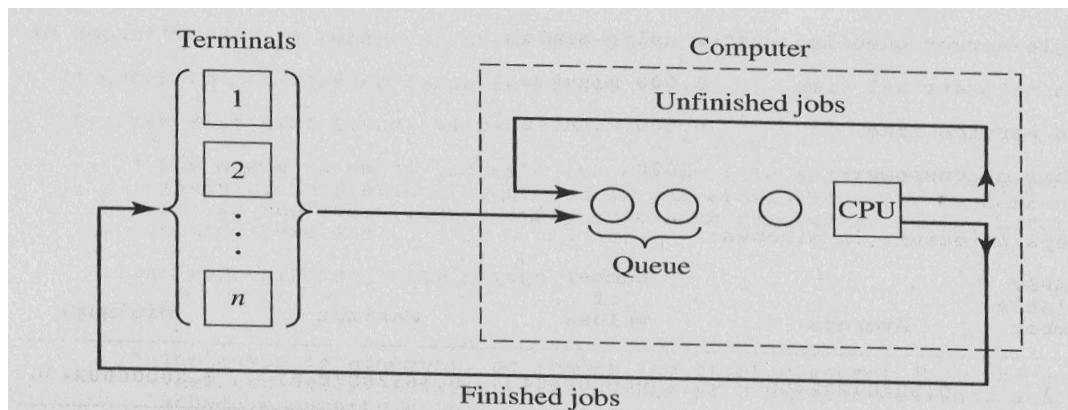


Aufgabe "optimales Time-Slicing in einer Server CPU" (5 Punkte)

Betrachten Sie folgendes Problem.

An n (default: $n=10$) Terminals sitzen Benutzer und führen unabhängig voneinander Aufgaben durch. Dabei wird jeweils nach einer "Denkzeit" von im Mittel 25 Sekunden (zufällig exponentialverteilt) Jobs an einen Server Rechner gesendet. Wir abstrahieren seine Rechenkapazitäten zu einer einzigen CPU die im Zeitscheibenverfahren (time-slice round-robin) einem Job eine maximale Rechenzeit (z.B. 0.1 Sekunden) zuteilt: die Zeitscheibendauer. Ist der Job innerhalb



dieser Zeitspanne nicht beendet worden, muss er wieder in die Warteschlange und eine weitere Zeitscheibe abwarten. Zeitscheiben werden immer ganz ausgeführt, selbst wenn der Job früher fertig wird (!). Jeder Wechsel eines Jobs in der CPU bringt einen Swapping Overhead von 0.015 Sekunden mit sich. Jobs haben einen zufällig verteilten Bedarf an CPU-Zeit (im Mittel 0.8 Sekunden, exponentialverteilt). Wenn der Job auf dem Server fertig ist, kehrt er wieder zu seinem Ausgangsterminal zurück mit dem Resultat. Hier denkt der Benutzer wieder nach und sendet dann den nächsten Job. Wir nehmen an, es gäbe kein natürliches Ende dieser Zyklen.

a) Modellieren Sie dieses System in AnyLogic. Die Zahl der Terminals soll als Parameter im Experiment einstellbar sein. Die Zeitscheibendauer und der Swapping Overhead sollen Modellparameter sein. Auch die mittlere Denkzeit und der mittlere Bedarf an CPU-Zeit eines Jobs sollen Parameter sein, und zwar potentiell für jedes Terminal verschieden. Terminals und Server sollen jeweils in einem eigenen ActiveObject/Agent modelliert werden. Terminals sind dann im Modell ein "replicated object".

Tipp: Sie müssen die Rückverbindung vom Server zum Terminal über einen Exit-Block im ersten und einen Enter-Block im letzteren modellieren. Die Hinverbindung hingegen kann man mit einem normalen Connector realisieren.

Simulieren Sie das System mit den oben angegebenen Defaultwerten, und beobachten Sie anhand geeigneter Statistiken und Plots das Systemverhalten. Beachten Sie insbesondere, ab wann das System "visuell" einen Steady State erreicht hat.

b) Definieren Sie eine Zielfunktion: die mittlere reine Wartezeit des Terminalbenutzers, die als Wartezeit zwischen Absenden und Empfangen eines Jobs minus der benötigten CPU Rechenzeit definiert ist, und davon den über alle Terminals gemittelten Wert. Beobachten Sie die Abhängigkeit dieser Zielfunktion von der Zeitscheibendauer in einem Parametervariationsexperiment mit geeigneten Plots. Benutzen Sie dafür Replikationen, um

statistisch belastbare Aussagen zu bekommen. Variieren Sie in weiteren Variations-
experimenten die anderen Parameter. Versuchen dabei immer vorher eine qualitative
Prognose für das Resultat zu geben! Beschreiben und interpretieren Sie die
Korrektheit/Irrtum Ihrer Prognosen.

Tipp: deaktivieren Sie parallele Evaluierungen. Im Feld "after simulation" ist eine einzige Replikation abgelaufen, im Feld "after iteration" sind alle Replikationen einer einzigen Einstellung eines zu variierenden Parameters abgelaufen.

c) optimieren Sie für die Defaultwerte die Zeitscheibendauer mit der in b) angegebenen Zielfunktion in einem Optimierungsexperiment.

d) ist dieses Modell in etwa äquivalent zum Round Robin Scheduling wie in <https://www.cs.rutgers.edu/~pxk/416/notes/07-scheduling.html> beschrieben?

~~http://www.cs.tut.ac.uk/staff/~~
~~http://www.cs.tut.ac.uk/staff/~~

[illegible]