

Aufgabe 10:

SysProg Raytracer (sysray)

In dieser Aufgabe soll ein einfacher Raytracer so erweitert werden, dass dieser POSIX-Threads verwendet. Den Raytracer finden Sie im Verzeichnis *~i4sp/pub/Aufgabe10*. Kopieren Sie die Dateien in Ihr Arbeitsverzeichnis.

a) Beispiel anschauen

In der Datei *example.c* finden Sie ein Beispiel für die Benutzung von den Modulen *display.o* und *trace.o*. Benennen Sie die Datei in *sysray.c* um und passen Sie das *Makefile* entsprechend an. Nachdem Sie das Programm übersetzt und gestartet haben, sollte es das Bild (*img.jpeg*) berechnen und nach jeweils 10 Zeilen in einem Fenster darstellen. Das Fenster können Sie wieder schliessen in dem Sie die Taste 'q' im Fenster drücken.

Das Programm funktioniert wie folgt. In der Funktion **ray_shade** wird für den Punkt (x,y) der Farbwert bestimmt. Die Funktion **draw_pixel** zeichnet anschliessend an die Position (x,y) einen Punkt mit dem gespeicherten Farbwert. Mit der Funktion **update_window** wird die Darstellung im Fenster erneuert.

b) Um POSIX-Threads erweitern

Verändern Sie nun das Programm insoweit, dass wahlweise eine unterschiedliche Anzahl von Threads (**pthread_create**) das Bild berechnen. Die Anzahl der Threads sollen dem Programm als Argument übergeben werden. Das Bild soll nun wie folgt berechnet werden. Jeder erzeugte Thread nimmt einen der noch nicht berechneten Punkte und berechnet diesen, ist der Thread fertig nimmt er sich einen neuen Punkt zur Bearbeitung vor und so weiter. Das Bild soll weiterhin nach 10 berechneten Zeilen dargestellt werden und nachdem alle Threads sich beendet haben.

c) Messungen

Multithreading mit mehr als zwei Threads bringt erst Vorteile, wenn das Programm auch auf einem Rechner mit mehr als einem Prozessor läuft. In der Woche vom **23.1.** bis zum **30.1.** werden wir für die Übungsteilnehmer auf dem Rechner *kenny.informatik.uni-erlangen.de* Logins einrichten. Es handelt sich dabei um ein Dual-Xeon-Rechner, der unter Linux als Betriebssystem läuft. Machen Sie auf diesem Rechner Messungen mit 1, 2, 3, 4, 5 und 6 Threads. Interpretieren Sie die Ergebnisse und geben Sie Ihre Interpretation zusammen mit den Ergebnissen als *doc/sysray.txt* ab.

Hinweise:

Achten Sie darauf, dass Sie bei gemeinsam genutzt Ressourcen ausreichend synchronisieren (**pthread_mutex_lock**, **pthread_mutex_unlock**) müssen.

Mit der Funktion **ray_shade_nxn** kann die Darstellungsqualität durch Oversampling verbessert werden. Der Rechenaufwand verdoppelt sich dabei je Stufe. Wenn Sie diese Funktion verwenden, so geben Sie dieses bei den Messungen mit an.

Das Zeichnen im Fenster kann unter gewissen Umständen länger dauern als die Berechnung. Wenn Sie zum Beispiel eine X11-Verbindung über einen ssh-Tunnel benutzen.

Abgabe: bis spätestens Donnerstag, 30.01.2003, 15:30 Uhr