Personen: Alan Romer(803551),Aaron Köhler(803499)

Aufgabe 3.1

c)Probleme, welche zu bewältigen waren, das Timing, da ich zuerst den zweiten Prozess erstellen musste und dieser seine PID in die Pipe schreiben muss, bevor der erste Prozess draus lesen möchte. Dies habe ich durch ein sleep gelöst.

Aufgabe 3.2

Da ich mir nicht sicher bin was für Gantt Diagramme sie annehmen habe ich mir eine Definition aus dem Internet gesucht auf welches mein Gantt Diagramm zutrifft. Der Grund weshalb ich es so strukturiert habe ist da es meiner Meinung nach übersichtlicher ist als das welches wir als Beispiel in den Vorlesungen erhalten haben.

Def:  Links des Diagramms befindet sich eine Liste der Aktivitäten, auf einer Leiste oben – von links nach rechts – befindet sich eine passende Zeitleiste. Jede Aktivität wird durch einen Balken abgebildet, wobei dieser die das Startdatum, die Dauer und das Enddatum repräsentiert.

a)

x- Rechnend

SJF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | | 16 | | 17 | | 18 | | 19 | | 20 | | 21 | | 22 |
| P1 | x | | x | | x | | x | | x | | x | | x | | x | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | |
| P2 | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | x | | x | | x | | x | | x | |
| P3 | - | | - | | - | | - | | - | | - | | - | | - | | x | | x | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | |
| P4 | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | x | | x | | x | | x | | - | | - | | - | | - | | - | |
| P5 | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | x | | x | | x | | - | | - | | - | | - | | - | | - | | - | | - | | - | |

Round Robin

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | | 16 | | 17 | | 18 | | 19 | | 20 | | 21 | | 22 | |
| P1 | x | | x | | x | | - | | - | | x | | - | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | | x | |
| P2 | - | | - | | - | | x | | - | | - | | x | | - | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | |
| P3 | - | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | |
| P4 | - | | - | | - | | - | | - | | - | | - | | x | | - | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | | - | | x | | - | |
| P5 | - | | - | | - | | - | | - | | - | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | | - | | x | | - | | - | | - | | - | |

b)

SJF

W1=0 W2=14 W3=5 W4=8 W5=4

Mittlere Wartezeit=31/5

V1=8 V2=19 V3=7 V4=12 V5=7

Mittlere Verweilzeit=53/5

Round Robin

W1=14 W2=12 W3=4 W4=12 W5=9

Mittlere Wartezeit=51/5

V1=22 V2=17 V3=6 V4=16 V5=12

Mittlere Verweilzeit=73/5

Aufgabe3.3

Orange für die Zelle bevor sie ankommt

Rot für die Rechenzeit

Blau für die Zelle bevor der Prozess beendet sein soll

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | |
| P1 | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | x | | x | | - | | - | | - | |
| P2 | - | | - | | x | | x | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | | - | |
| P3 | - | | - | | - | | - | | x | | x | | - | | - | | - | | - | | - | | - | | - | | - | | - | |
| P4 | - | | - | | - | | - | | - | | - | | - | | - | | x | | x | | - | | - | | - | | - | | - | |
| P5 | - | | - | | - | | - | | - | | - | | x | | x | | - | | - | | - | | - | | - | | - | | - | |

Def aus der Vorlesung

Ein Vergabeplan für eine Prozess menge P = {P1, P2, . . . , Pn} mitgegebenen ri,ti,di heißt gültig, falls sich keine Ausführungszeiten auf einem Prozessor überlappen und sämtliche Zeitbedingungen eingehalten werden.

Zeitbedingung ri ≤ si ∧ si + ti ≤ di

Ein Beispiel eines gültigen Ablaufplans

Aufgabe 3.4

Windows 2000 Zeitscheiben länge aus der Vorlesung: 20 ms

a)

i) Priority boost nach i/o -> Zeitscheibe: 20 ms

ii) Priority boost von Netzwerk: 2 -> nach 40 ms ist es wieder bei ihrer Basispriorität (siehe Aufgabe iii)

iii)in meiner seperaten Priorität 8 Warteschlange steht P1 vor P3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 ms | | 20 | | 40 | | 60 | | 80 | | 100 | | 120 | | 140 | | 160 | | 180 | | 200 | | 220 | | 240 | | 260 | |
| P1 | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | |
| P2 | 10 | | 9 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | |
| P3 | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | |

b)

i) Priority boost nach GUI-> eine Scheibe ist 3-mal so lang wie eine normale zeitscheibe daher: 60 ms

ii) für eine zeitscheibe um 6 -> auf 14 geboostet nach 60 ms hat es wieder seine Basisprirität

iii) in meiner seperaten Priorität 8 Warteschlange steht P1 vor P3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | | 20 | | 40 | | 60 | | 80 | | 100 | | 120 | | 140 | | 160 | | 180 | | 200 | | 220 | | 240 | | 260 | | 280 | |
| P1 | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | |
| P2 | 14 | | 14 | | 14 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | |
| P3 | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | | 8 | |

c)

i) Starvation Vermeidung: 20 ms ist eine Zeitscheibe und neu prio wird von 8 auf 15 gesetzt -> um 7 erhöht

ii) Starvation Vermeidung boosting für 2 Zeitscheiben -> nach 40 ms ist es wieder bei ihrer Basispriorität

iii) Dem verhungern von Prozessen wird in hauptsächlich 3 wegen entgegengewirkt.

1. Der Baum nimmt immer den task ganz links für bis dieser nicht mehr die geringste v runtime hat aber mindestens 10 ms. Dadurch wird im Prozessor versucht das alle Task bzw. Task gruppen eine gleiche bzw. ähnliche vruntime haben.
2. Task grouping damit ein Task nicht sehr viele Task erzeugen kann und somit den Prozessor alleine für sich beansprucht
3. Da die bisher verbrauchte CPU-Zeit in die Berechnung der vruntime mit eingeht, verschlechtern sich Prozesse, die rechnen.

Ein weg, welcher funktionieren sollte, um einen Prozess zu verhungern, ist indem wir einen Prozess haben mit welcher für eine sehr lange Zeit alleine rechnet. Aufgrund des Wissens wie vruntime berechnet wird: Laufzeit /Anzahl Prozesse gewichtet mit Priorität. Wodurch vruntime sich schneller erhöht für Prozesse niedriger Priorität. Dadurch erzielen wir einen Prozess mit einer großen vruntime. Wenn wir dann davon ausgehen, dass mehrere weitere langläufige Prozesse hinzukommen in regelmäßigen Abständen. So musste es unmöglich sein für den Baum die runtime aller Prozesse auf ein Niveau mit dem vorher lange rechnenden Prozess zu bringen bevor neu Prozesse kommen. Daher wäre es theoretisch möglich einen Prozess zu verhungern. Dies wurde aber in der Vorlesung erwähnt und als fork bombe bezeichnet, da es nicht möglich sein soll so viele Prozesse hinzuzufügen, um ihn verhungern zu lassen.

Da bei einer fork bombe nichts mehr funktioniert sollte es daher im Normalfall nicht möglich sein ein Prozess zu verhungern.

Aufgabe 3.5

c)

Mein Rechner

Architektur: i386

Betriebssystem: Darwin Kernel Version 19.6.0: Thu Jun 18 20:49:00 PDT 2020; root:xnu-6153.141.1~1/RELEASE\_X86\_64

Geschätzte Stäckgröße:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

gpu1

Architektur: unknown

Betriebssystem: #1 SMP Debian 4.9.240-2 (2020-10-30) 