Betriebssysteme HA Mattis Best 35213981

Aufgabe 1)

(a)

1)

Der Algorithmus lässt alle Philosophen warten bis ihr Index als Ergebnis der Modulo Rechnung aus einem Runden Counter, der nach dem einer gegessen hat erhöht wird, und der Anzahl der Philosophen hervorgeht. Damit entstehen die Restklassen 0,1,2,3,4,5,6,7 was den Indexen der Philosophen entspricht. Somit essen keine Philosophen gleichzeitig und es entsteht ein Kreislauf, da 8 ≡ 0 mod 8 ist und 9 ≡ 1 mod 8 ist etc.

2)

Der Algorithmus ist fair, da jeder Philosoph etwa gleich viel zu essen bekommt, da durch die Reihenfolge fest ist das alle Philosophen drankommen und nach dem letzten wieder der Erste essen kann, wodurch auch gegeben ist das kein Philosoph verhungern wird. Das einzige Problem ist das die Ressourcen nicht optimal genutzt werden, da immer nur einer essen kann und der Rest warten muss.

(b)

1)

Der Algorithmus gibt an das alle Philosophen erst die linke und dann die rechte Gable besetzen außer der Philosoph mit dem Index == 3, welcher erst die rechte und dann die linke Gable besetzt. Somit ist der Deadlock ausgeschlossen, wenn alle gleichzeitig die linke Gabel nehmen und alles warten bis die jeweils rechte freigegeben wird.

Nimmt die 3 seine Rechte Gable vor der 2 kann die drei essen und daraufhin kann dann die vierte essen was einen Kreislauf nach sich zieht, bei dem nachdem die 5 fertig ist mit essen die vier wieder essen kann weil die 3 auf das auf nehmen seiner rechten Gable wartet und somit die linke Gabel der 3 noch offen ist. Also können in diesem Algorithmus auch mehrere Philosophen gleichzeitig essen.

Nimmt die 2 die linke zuerst muss die 3 warten bis die 2 die Gable wieder frei gibt wodurch die linke der 3 frei ist und die 4 als erstes isst und dann wieder derselbe Kreislauf entsteht.

2)

Der Algorithmus ist fair, da alle Philosophen ungefähr gleich oft essen können somit muss auch keiner verhungern. In diesem Fall sind die Ressourcen auch recht gut ausgenutzt worden, da immer zwei essen bis auf am Anfang wo erst nur einer isst und der zweite später beginnt.

Aufgabe 3)

(1)

Path(A ; B), D end

< P(s1) ( A ; B ) V(s1) > < P(s1) D V(s1) >

< P(s1) A V(s2) P(s2) B V(s1) > < P(s1) D V(s1) >

(2)

Path 1:([A] , [B]) end

< P(s1) [A] V(s1) > < P(s1) [B] V(s1) >

< PP(c, s2, P(s1)) A VV(c, s2 , V(s1)) > < PP(c, s3, P(s1)) B VV(c, s3 , V(s1)) >

P(s2) P(s2) P(s3) P(s3)

c = c + 1 c = c - 1 c = c + 1 c = c - 1

If c = 1 then P(s1) if c = 0 then V(s1) if c = 1 then P(s1) if c = 0 then V(s1)

V(s2) V(s2) V(s3) V(s3)