CS 520

SR Assignment 1

Fall 2015

Christian White Masc0870

**Script of program being run**

Script started on Wed Nov 25 13:20:12 2015

> ^G^G^Gsr prog1.sr^M^M

> a.out^M^M

Four children have 10 quarters each

they inefficiently run back and fourth throwing 1 quarter in a bucket each time

child 1 has added a coin

there are now 1 and a total of $ 0.250000 dollars

child 2 has added a coin

there are now 2 and a total of $ 0.500000 dollars

child 3 has added a coin

there are now 3 and a total of $ 0.750000 dollars

child 4 has added a coin

there are now 4 and a total of $ 1.00000 dollars

child 1 has added a coin

there are now 5 and a total of $ 1.25000 dollars

child 3 has added a coin

there are now 6 and a total of $ 1.50000 dollars

child 4 has added a coin

there are now 7 and a total of $ 1.75000 dollars

child 2 has added a coin

there are now 8 and a total of $ 2.00000 dollars

child 1 has added a coin

there are now 9 and a total of $ 2.25000 dollars

child 2 has added a coin

there are now 10 and a total of $ 2.50000 dollars

child 4 has added a coin

there are now 11 and a total of $ 2.75000 dollars

child 3 has added a coin

there are now 12 and a total of $ 3.00000 dollars

child 1 has added a coin

there are now 13 and a total of $ 3.25000 dollars

child 2 has added a coin

there are now 14 and a total of $ 3.50000 dollars

child 3 has added a coin

there are now 15 and a total of $ 3.75000 dollars

child 4 has added a coin

there are now 16 and a total of $ 4.00000 dollars

child 1 has added a coin

there are now 17 and a total of $ 4.25000 dollars

child 3 has added a coin

there are now 18 and a total of $ 4.50000 dollars

child 2 has added a coin

there are now 19 and a total of $ 4.75000 dollars

child 4 has added a coin

there are now 20 and a total of $ 5.00000 dollars

child 1 has added a coin

there are now 21 and a total of $ 5.25000 dollars

child 2 has added a coin

there are now 22 and a total of $ 5.50000 dollars

child 3 has added a coin

there are now 23 and a total of $ 5.75000 dollars

child 4 has added a coin

there are now 24 and a total of $ 6.00000 dollars

child 1 has added a coin

there are now 25 and a total of $ 6.25000 dollars

child 2 has added a coin

there are now 26 and a total of $ 6.50000 dollars

child 4 has added a coin

there are now 27 and a total of $ 6.75000 dollars

child 3 has added a coin

there are now 28 and a total of $ 7.00000 dollars

child 2 has added a coin

there are now 29 and a total of $ 7.25000 dollars

child 3 has added a coin

there are now 30 and a total of $ 7.50000 dollars

child 4 has added a coin

there are now 31 and a total of $ 7.75000 dollars

child 1 has added a coin

there are now 32 and a total of $ 8.00000 dollars

child 2 has added a coin

there are now 33 and a total of $ 8.25000 dollars

child 4 has added a coin

there are now 34 and a total of $ 8.50000 dollars

child 1 has added a coin

there are now 35 and a total of $ 8.75000 dollars

child 3 has added a coin

there are now 36 and a total of $ 9.00000 dollars

child 2 has added nine coins and is waiting for the others to finish

child 1 has added nine coins and is waiting for the others to finish

child 3 has added nine coins and is waiting for the others to finish

child 4 has added nine coins and is waiting for the others to finish

all children have added 9 of there coins

child 4 has added a coin

there are now 37 and a total of $ 9.25000 dollars

child 1 has added a coin

there are now 38 and a total of $ 9.50000 dollars

child 2 has added a coin

there are now 39 and a total of $ 9.75000 dollars

child 3 has added a coin

there are now 40 and a total of $ 10.0000 dollars

> ^D^H^Hexit

script done on Wed Nov 25 13:20:45 2015

**Second run with more time added to nap**

#added to show stronger evidence of behavior

> sr prog1\_a.sr

> a.out

Four children have 10 quarters each

they inefficiently run back and fourth throwing 1 quarter in a bucket each time

child 1 has added a coin

there are now 1 and a total of $ 0.250000 dollars

child 2 has added a coin

there are now 2 and a total of $ 0.500000 dollars

child 3 has added a coin

there are now 3 and a total of $ 0.750000 dollars

child 4 has added a coin

there are now 4 and a total of $ 1.00000 dollars

child 1 has added a coin

there are now 5 and a total of $ 1.25000 dollars

child 2 has added a coin

there are now 6 and a total of $ 1.50000 dollars

child 3 has added a coin

there are now 7 and a total of $ 1.75000 dollars

child 4 has added a coin

there are now 8 and a total of $ 2.00000 dollars

child 1 has added a coin

there are now 9 and a total of $ 2.25000 dollars

child 3 has added a coin

there are now 10 and a total of $ 2.50000 dollars

child 2 has added a coin

there are now 11 and a total of $ 2.75000 dollars

child 4 has added a coin

there are now 12 and a total of $ 3.00000 dollars

child 1 has added a coin

there are now 13 and a total of $ 3.25000 dollars

child 3 has added a coin

there are now 14 and a total of $ 3.50000 dollars

child 2 has added a coin

there are now 15 and a total of $ 3.75000 dollars

child 4 has added a coin

there are now 16 and a total of $ 4.00000 dollars

child 3 has added a coin

there are now 17 and a total of $ 4.25000 dollars

child 2 has added a coin

there are now 18 and a total of $ 4.50000 dollars

child 1 has added a coin

there are now 19 and a total of $ 4.75000 dollars

child 3 has added a coin

there are now 20 and a total of $ 5.00000 dollars

child 4 has added a coin

there are now 21 and a total of $ 5.25000 dollars

child 2 has added a coin

there are now 22 and a total of $ 5.50000 dollars

child 1 has added a coin

there are now 23 and a total of $ 5.75000 dollars

child 2 has added a coin

there are now 24 and a total of $ 6.00000 dollars

child 3 has added a coin

there are now 25 and a total of $ 6.25000 dollars

child 4 has added a coin

there are now 26 and a total of $ 6.50000 dollars

child 1 has added a coin

there are now 27 and a total of $ 6.75000 dollars

child 3 has added a coin

there are now 28 and a total of $ 7.00000 dollars

child 2 has added a coin

there are now 29 and a total of $ 7.25000 dollars

child 4 has added a coin

there are now 30 and a total of $ 7.50000 dollars

child 1 has added a coin

there are now 31 and a total of $ 7.75000 dollars

child 3 has added a coin

there are now 32 and a total of $ 8.00000 dollars

child 2 has added a coin

there are now 33 and a total of $ 8.25000 dollars

child 3 has added nine coins and is waiting for the others to finish

child 4 has added a coin

there are now 34 and a total of $ 8.50000 dollars

child 1 has added a coin

there are now 35 and a total of $ 8.75000 dollars

child 2 has added nine coins and is waiting for the others to finish

child 4 has added a coin

there are now 36 and a total of $ 9.00000 dollars

child 1 has added nine coins and is waiting for the others to finish

child 4 has added nine coins and is waiting for the others to finish

all children have added 9 of there coins

child 4 has added a coin

there are now 37 and a total of $ 9.25000 dollars

child 1 has added a coin

there are now 38 and a total of $ 9.50000 dollars

child 2 has added a coin

there are now 39 and a total of $ 9.75000 dollars

child 3 has added a coin

there are now 40 and a total of $ 10.0000 dollars

**Code implementation**

resource barrier()

#number of processes

const N := 4

#counter to signal last process

var count := 0

var coins := 0

var total: real := 0

var ct: int

var x: real

#continue allows for all processes to play, and all initially set to 0.

sem mutex := 1, continue[N] := ([N] 0)

write("Four children have 10 quarters each")

write("they inefficiently run back and fourth throwing 1 quarter in a bucket each time")

#Creates 4 children(workers)

process worker(i := 1 to N)

fa j := 1 to 9 ->

coins := coins + 1;

total := coins \* 0.25;

write("child ", i ," has added a coin");

write("there are now ", coins ," and a total of $", total," dollars");

#adds randomization to processes

#randomization code was adapted to this program from the boundedBuffer.sr program

x := random();

ct:= int(x\*50 + 50);

nap(ct)

af

#enter critical section

P(mutex)

write("child ", i ," has added nine coins and is waiting for the others to finish")

count := count + 1

#exit critical section

V(mutex)

#if all proccesse have completed there work then all can continue to do the last job.

if count = 4 ->

write("all children have added 9 of there coins")

fa w := 1 to N -> V(continue[w]) af

fi

#delay point, all processes wait here until allowed to move on.

P(continue[i])

coins := coins + 1;

total := coins \* 0.25;

write("child ", i ," has added a coin");

write("there are now ", coins ," and a total of $", total," dollars");

end

end