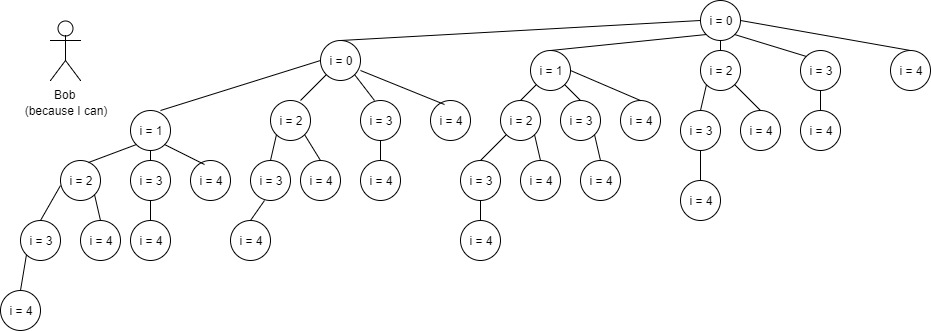
4. c)



i < 5 = 32 processes; i < 6 = 64 processes; i < 7 = 128 processes

7. a) Writer 1 Enters the database; ActiveWriters == 1

b) Reader 1 blocks on P(OkToRead); WaitingReaders == 1

c) Reader 2 blocks on P(OkToRead); WaitingReaders == 2

d) Writer 2 blocks on P(OkToWrite); WaitingWriters == 1

e) Reader 3 blocks on P(OkToRead); WaitingReaders == 3

f) Writer 1 leaves; ActiveWriters == 0;

Checks WaitingWriters and then V(OkToWrite);

Writer 2 enters the database; ActiveWriters == 1; WaitingWriters == 0

g) Reader4 blocks on P(OkToRead); WaitingReaders == 4

h) Writer 1 leaves; ActiveWriters == 0; Checks WaitingWriters

Enters while loop to wake up Readers. Readers 1, 2, 3 and 4 enter the database

WaitingReaders == 0; ActiveReaders == 4;

i) Writer 3 blocks on P(OkToWrite); WaitingWriters == 1;

Reader 5 blocks on P(OkToRead); WaitingReaders == 1;

Readers 1, 2, 3, 4 eventually leave the database; ActiveReaders == 0;

V(OkToWrite); Writer 3 enters the database; WaitingWriters == 0; ActiveWriters == 1

Writer 3 leaves; ActiveWriters == 0; Checks WaitingWriters

Enters while loop to wake up Reader 5; Reader 5 enters the database; ActiveReaders == 1;

WaitingReaders == 0; Reader 5 leaves the database; ActiveReaders == 0

12. b) 1st solution –

global:

plantingDone = create\_mbox(1);

// gardener mailboxes

seeds = create\_mbox(1);

soil = create\_mbox(1);

manure = create\_mbox(1);

process Vendor:

while (1) {

switch (random\*(3) { /\*pick random gardener\*/

case 1:

put soil and manure on the table

Send (seeds); /\*wake up gardener Seed\*/

break;

case 2:

put seeds and manure on the table

Send(soil); /\*wake up gardener Soil\*/

break;

case 3:

put seeds and soil on the table

Send(manure); /\* wake up gardener Manure\*/

break;

}

Receive(plantingDone); /\*wait for gardener to finish\*/

}

process seeds:

while(1) {

Receive(seeds);

get soil & manure;

do a planting;

Send(plantingDone);

}

process soil:

while(1) {

Receive(soil);

get seeds and manure;

do a planting;

Send(plantingDone);

}

process manure:

while(1) {

Receive(manure);

get soil and seeds;

do a planting;

Send(plantingDone);

}

2nd solution –

globals:

OkToPlant\_mutex = create\_mbox(1, 0);

crit\_section = create\_mbox(1, 0);

seed = create\_mbox(2, 1);

soil = create\_mbox(2, 1);

manure = create\_mbox(2, 1);

process Vendor:

while (1) {

switch(random\*(3)) { /\* select gardener \*/

case 1:

Send(crit\_section); // to prevent time slicing between sends

Send(seed, soil);

Send(seed, manure);

Receive(crit\_section);

break;

case 2:

Send(crit\_section);

Send(soil, seeds);

Send(soil, manure);

Receive(crit\_section);

break;

case 3:

Send(crit\_section);

Send(manure, seeds);

Send(manure, soil);

Receive(crit\_section);

break;

}

Receive(OkToPlant\_mutex);

}

process Seeds:

while(1) {

Send(OkToPlant\_mutex);

Receive(seed, soil);

Receive(seed, manure);

<do a planting…>

Receive(OkToPlant\_mutex);

}

process Soil:

while(1) {

Send(OkToPlant\_mutex);

Receive(soil, seeds);

Receive(soil, manure);

<do a planting…>

Receive(OkToPlant\_mutex);

}

process Manure:

while(1) {

Send(OkToPlant\_mutex);

Receive(manure, seeds);

Receive(manure, soil);

<do a planting…>

Receive(OkToPlant\_mutex);

}