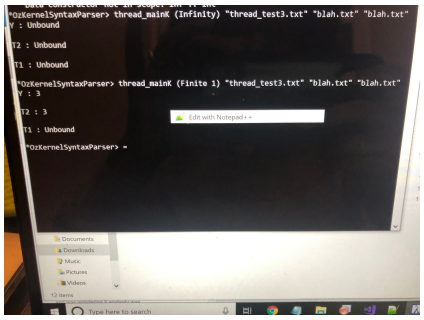
Madhuri Pyreddy
8 October 2019

CSCI 117 Lab 7

Part One

The possible thread sequences:

1. S1 T1 S2 S3 S3.1 T2 -- Display True
   S1, S2, T2, S3, T1 -- Display Error
   S1, S2, T1, S3, S3.1 T2 -- Display True
   S1 T1 S2 T2 -- Display Error
   S1 S2 T1 T2 -- Display Error
   S1 S2 T2 T1 -- Display Error
2. 
   
   The reason the outputs Y, T2, and T1 are unbound because when the main thread is complete, it doesn't execute those values since it switches back to the main thread for binding. When I ran quantum 1, the output ran Y:3, T2:3, and T1: unbound. The reason is that it is converted from syntax sugar to kernel syntax. Because of the kernel syntax (4+3), it couldn't compute T1 and it switched back to the main thread to complete the program before completing the binding to compute T1.
3. local Z in
   Z = 3
   thread local X in
   X = 1
   skip Browse X
   skip Browse X
   skip Basic
   skip Browse X
   skip Browse X
   skip Basic
   skip Browse X
   end
   end
   thread local Y in
   Y = 2
   skip Browse Y
   skip Basic

```
  skip Browse Y
  skip Browse Y
  skip Basic
  skip Browse Y
  skip Browse Y
  end
 end
 skip Browse Z
 skip Browse Z
 skip Browse Z
 skip Basic
 skip Browse Z
 skip Browse Z
end
```

4. The minimum quantum that will cause a suspension to occur is 5. The result is not what I expected is because the suspension occurred due to the syntax sugar from number 4 is converted to kernel syntax.
5. 
    a. fib1_sugar.txt

| | | | |
|---|---|---|---|
| X = 8 | 0.08s | 31,164,288 bytes | 34 |
| X = 10 | 0.36s | 137,178,944 bytes | 89 |
| X = 12 | 1.95s | 804,892,728 bytes | 233 |
| X = 13 | 6.02s | 2,034,900,224 bytes | 377 |

fib1_thread.txt

| | | | |
|---|---|---|---|
| X = 12 | 7.94s | 2,410,342,456 bytes | 144 |
| X = 13 | 29.67s | 6,221,077,712 bytes | 377 |
| X = 14 | 67.80 s | 16,110,732,168 bytes | 610 |
| X = 15 | 162.51 s | 41,801,603,520 bytes | 987 |

fib2_sugar.txt

| | | | |
|---|---|---|---|
| X = 13 | 0.03s | 11,870,632 bytes | 233 |
| X = 15 | 0.03s | 12,510,760 bytes | 610 |
| X = 17 | 0.04s | 13,201,776 bytes | 1597 |
| X = 19 | 0.04s | 13,941,360 bytes | 4181 |

For the fib1_sugar.txt, the fastest time was 0.08 seconds at X = 8. For the fib1_sugar.txt, the slowest time was 6.02s at X = 13. For the fib2_sugar.txt, the fastest time was 0.03s at X = 13. For the fib2_sugar.txt, the slowest time was 0.04s at X = 19. For the fib1_thread, the fastest time was 7.94 s at X = 12. For the fib1_thread, the slowest time was 162.51s at X = 15. The reason the time is slower in the fib1_thread is because it is switching back to the main thread to complete the program before completing the binding.

b. The pattern for the (n-1)+(n-2)+2 for n >= 2. For 0 and 1, there are 0 threads because there are no threads. For 2, there are 2 threads because there are 0 threads in 0 and 0 threads in 1 so, by using the formula, we do 0+0+2 = 2. For 3, there are a number of threads in 2 plus the number of threads in 1. So for 3, it would be 2 threads in 2 and 0 threads in 1. So, 2+0+2 = 4 threads. For 4, it would be a number of threads in 3 plus the number of threads in 2. So,it would be 4+2+2 = 8 threads. For 5, it would be a number of threads of 4 plus the number of threads from 3. So, it would be 8+4+2= 14 threads.

Part 2

1. 
```
local Producer Consumer OddFilter Filter N L P F Add in
   Producer = proc {$ N Limit Out}
  if (N<Limit) then T N1 in
    Out = (N|T)
    N1 = (N + 1)
    {Producer N1 Limit T}
  else Out = nil
  end
end
 OddFilter = proc {$ P Out}
   Filter = fun {$ O1 T1}
      case O1 of nil then T1
         [] '|'(1:H 2:T) then S in
```

```
                if ((H mod 2) == 1) then
                    S = {Filter T T1}
                      S
                      else
                        S = {Filter T T1}
                        (H|S)
                      end
                    end
                  end
                  Out = {Filter P nil}
                end
```

2. 
```
Consumer = fun {$ P}
    case P of nil then 0
        [] '|'(1:H 2:T) then
           (H + {Consumer T})
        end
end
```

3. 
```
//Example Testing
  N = 0
  L = 100
  thread {Producer N L P}
     skip Browse P
  end
  thread {OddFilter P F}
     skip Browse F
   end
  thread Add = {Consumer F}
      skip Browse Add
  end
      skip Browse P
      skip Browse F
      skip Browse Add
end
```