CS 101: Computer Programming and Utilization

06-Muddy Points & Revision

Instructor: Sridhar Iyer
IIT Bombay
## Muddy Points

### Questions (AM batch)

<table>
<thead>
<tr>
<th></th>
<th>Basic</th>
<th>Advanced</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threads &amp; Sync</td>
<td>42</td>
<td>26</td>
<td>68</td>
</tr>
<tr>
<td>Arrays &amp; Variables</td>
<td>12</td>
<td>14</td>
<td>26</td>
</tr>
<tr>
<td>Scratch specific</td>
<td>11</td>
<td>4</td>
<td>15</td>
</tr>
</tbody>
</table>

### Questions (PM batch)

<table>
<thead>
<tr>
<th></th>
<th>Basic</th>
<th>Advanced</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threads &amp; Sync</td>
<td>17</td>
<td>8</td>
<td>25</td>
</tr>
<tr>
<td>Arrays &amp; Variables</td>
<td>10</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>Scratch specific</td>
<td>15</td>
<td>23</td>
<td>28</td>
</tr>
</tbody>
</table>

- Responses are uploaded on Moodle
Some purposes of variables –

- Allow same code to run on different inputs
- Keep track of iterations
- Do code-tracing (hand-execution) of the program shown when the value of b is input as 9.
Threads - Revision

- High-level program \(\rightarrow\) Assembly program \(\rightarrow\) Binary
- One instruction in high-level \(\rightarrow\) Many instructions in assembly; See Centigrade conversion program from class 01-intro
- Multiple threads \(\rightarrow\) The corresponding assembly programs share the CPU, turn-by-turn execution of a few instructions from each
- Program counter \(\rightarrow\) Keeps track of the instruction currently under execution, for each thread
- One instruction in assembly \(\rightarrow\) very small, cannot be interrupted; atomic

Diagrams on the next slide
Multiple Threads on a single CPU

Scratch
Thread 1
Statement 1
Statement 2
Statement 3
....

Assembly
Thread 1
Instruction 1.1
Instruction 1.2
Instruction 2.1
Instruction 2.2
Instruction 2.3

Not Atomic

Different instructions require different amounts of time

Atomic

CPU

1.1 a.1 1.2 b.1 2.1 b.2 T3

Interleaved Sequential Execution

Time

New Thread T3 starts → non-determinism

Fixed duration time-slice

Interleaved Sequential Execution

Not Atomic

Atomic
Threads: Think-Pair-Share Activity

Consider three threads as shown below.
Assume that: (i) 'When' and 'Say' statements result in 2 assembly instructions, and (ii) all other statements result in 3 assembly instructions. Also assume that: (a) assembly instructions are atomic and take the same amount of time, and (b) CPU time-slice is sufficient for 3 assembly instructions.
Show two possible interleaved execution sequences.

Thread A
When Run flag clicked,
Say “Thread A start”;
Broadcast “event”;
Move 10 steps;
Say “Thread A done”;

Thread B
When Run flag clicked
Say “Thread B start”;
Turn 90 degrees;
Say “Thread B done”;

Thread C
When I receive “event”,
Glide to (0,0);
Say “Thread C done”;
Threads Activity: Solution

Thread A
Assembly Instrns
When:- A1.1; A1.2
Say:- A2.1; A2.2
Bcast:- A3.1, A3.2, A3.3
Move: -A4.1, A4.2, A4.3
Say:-A5.1, A5.2

Thread B
Assembly Instrns
When:- B1.1; B1.2
Say: B2.1, B2.2
Turn:- B3.1, B3.2, B3.3
Say:- B4.1, B4.2

Thread C
Assembly Instrns
When:- C1.1, C1.2
Glide:- C2.1, C2.2, 2.3
Say:- C3.1, C3.2

CPU Execution Sequence 1:

Alternate Sequences: (i) C starts before B3.3, (ii) B starts before A
Thread Synchronization

Broadcast-Receive

Thread 1

Receive event

Broadcast event

Run: demo06-paddle.sb

Thread 2

Boolean-Variable

Thread 1

Set Flag = 1

Wait until <Flag = 1>

Thread 1

Flag

Initialized to 0

Thread 2

Run: demo06-QandA.sb
Announcement

This revision of Muddy Points on should help you prepare for Quiz1.

If you have more queries

- Post them on Moodle. TAs will answer.
- Talk to your Lab TA.

Friday, 31st Jan:

- Quiz1 – starts at 08:15 AM, ends at 09:25 AM.
- No class on that day – for both batches.