

IN3030 Oblig5 report - hermagst@uio.no

User guide

To run the program simply compile the single java class and run it. This also serves as the test program specified in the oblig. The class is mainly the same as the precode meaning it handles the same command line arguments (requires number of threads, but also optionally some other arguments).

```
javac *.java
java WaitNext2 $(nproc) # + any other arguments
```

Implementation

This solution only consists of some very minimal inlined changes to the precode. Some of the changes also makes large portions of the precode essentially dead code and could also maybe have some more debugging added to it, but this is not something I felt was worth changing, also because I felt the amount of debugging in the `waitNext` function was a little overwhelming. The main change areas to look for are:

- `first` -> `firstState`: Just a namechange to make more sense with the rest of the `waitAndSwap` solution.
- `finishOrder`: This serves the main testing function of my solution. Every time a thread finishes its execution it appends its id to this list and prints it out. To verify the solution works, you can look at the final output to stdout before the program halts and verify that it follows the pair-wise swapped pattern described in the oblig, e.g. "1, 0, 3, 2, 5, 4, 7" for 8 threads. Note that this list is 0 base indexed compared to 1 based as in the oblig. Not very sophisticated or automated, but for a small problem like this it's sufficient.
- `skipState`: Where the `waitNext` function effectively has two control paths it can go through (one for the first calling thread and one for the next calls), the `waitAndSwap` function has three paths.
 - For the first calling thread it follows the `firstState` control path which is the same as in `waitNext` with the only difference being that it sets the state of the class to `skipState`.
 - While in `skipState` we basically do nothing, clean up the semaphores and exit the `skipState`.
 - Finally if we're not in the `firstState` and not in the `skipState` we set the state to `skipState` and then follow the same control path as the second state in `waitNext`
- `Worker` class: I have changed the worker class to ignore the `N` variable and simply hardcoded it to only run one implementation, as this made it simpler to test.

Appendix

Following is the full command line output for running the program with 12 cores (`java WaitNext2 12`)

```
threads: 4
*****
*****
```

```

NOTE: after the first thread has entered HOLDING, there will always be ONE
thread either WAITING or about to WAIT, so the program will NEVER
terminate!
Number of threads: 12; iterations: 4; debug: 9; varispeed: 100 ms;
extra slow: 0
Thread 8, 0 START thread
Thread 10, 0 START thread
Thread 1, 0 START thread
Thread 9, 0 START thread
Thread 11, 0 START thread
Thread 3, 0 START thread
Thread 0, 0 START thread
Thread 7, 0 START thread
Thread 6, 0 START thread
Thread 5, 0 START thread
Thread 2, 0 START thread
Thread 4, 0 START thread
Thread 10, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 1, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 2, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 5, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 11, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 9, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 6, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 3, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 0, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 7, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 8, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 4, 0 Sems: KICK 1 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 0, 0 variSpeed delay: 85 ms
Thread 0, 0 Sems: KICK 0 Q: 0; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 0, 0 resuming after variSpeed delay
Thread 0, 0 Sems: KICK 0 Q: 8; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 0, 0 variSpeed delay: 78 ms
Thread 0, 0 Sems: KICK 0 Q: 8; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 0, 0 resuming after variSpeed delay
Thread 0, 0 Sems: KICK 0 Q: 11; ENTER 1 Q: 0; HOLD:0 Q: 0
Thread 0, 0 variSpeed delay: 51 ms
Thread 0, 0 Sems: KICK 0 Q: 11; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 0, 0 resuming after variSpeed delay
Thread 0, 0 Sems: KICK 0 Q: 11; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 1, 0 variSpeed delay: 14 ms
Thread 1, 0 Sems: KICK 0 Q: 10; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 1, 0 resuming after variSpeed delay
Thread 1, 0 Sems: KICK 0 Q: 10; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 1, 0 variSpeed delay: 11 ms
Thread 1, 0 Sems: KICK 0 Q: 10; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 1, 0 resuming after variSpeed delay
Thread 1, 0 Sems: KICK 0 Q: 10; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 1, 0 variSpeed delay: 69 ms
Thread 2, 0 variSpeed delay: 7 ms
Thread 1, 0 Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 2, 0 Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 2, 0 resuming after variSpeed delay

```

```
Thread 2, 0    Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 0, 0    variSpeed delay: 4 ms
Thread 2, 0    variSpeed delay: 57 ms
Thread 0, 0    Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 2, 0    Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 0, 0    resuming after variSpeed delay
Thread 0, 0    Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 0, 0    END thread
Thread 0, 0    Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 0
[0]
Thread 2, 0    resuming after variSpeed delay
Thread 2, 0    Sems: KICK 0 Q: 9; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 3, 0    variSpeed delay: 1 ms
Thread 3, 0    Sems: KICK 0 Q: 8; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 3, 0    resuming after variSpeed delay
Thread 3, 0    Sems: KICK 0 Q: 8; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 3, 0    variSpeed delay: 5 ms
Thread 3, 0    Sems: KICK 0 Q: 8; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 1, 0    resuming after variSpeed delay
Thread 1, 0    Sems: KICK 0 Q: 8; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 1, 0    END thread
Thread 1, 0    Sems: KICK 0 Q: 8; ENTER 0 Q: 0; HOLD:0 Q: 1
[0, 1]
Thread 3, 0    resuming after variSpeed delay
Thread 3, 0    Sems: KICK 0 Q: 8; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 3, 0    variSpeed delay: 24 ms
Thread 4, 0    variSpeed delay: 86 ms
Thread 3, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 4, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 3, 0    resuming after variSpeed delay
Thread 3, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 3, 0    END thread
Thread 3, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 1
[0, 1, 3]
Thread 4, 0    resuming after variSpeed delay
Thread 4, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 2, 0    variSpeed delay: 76 ms
Thread 4, 0    variSpeed delay: 52 ms
Thread 2, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 4, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 4, 0    resuming after variSpeed delay
Thread 4, 0    Sems: KICK 0 Q: 7; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 5, 0    variSpeed delay: 10 ms
Thread 5, 0    Sems: KICK 0 Q: 6; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 5, 0    resuming after variSpeed delay
Thread 5, 0    Sems: KICK 0 Q: 6; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 5, 0    variSpeed delay: 27 ms
Thread 5, 0    Sems: KICK 0 Q: 6; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 2, 0    resuming after variSpeed delay
Thread 2, 0    Sems: KICK 0 Q: 6; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 2, 0    END thread
Thread 2, 0    Sems: KICK 0 Q: 6; ENTER 0 Q: 0; HOLD:0 Q: 1
[0, 1, 3, 2]
Thread 5, 0    resuming after variSpeed delay
```

```
Thread 5, 0    Sems: KICK 0 Q: 6; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 5, 0    variSpeed delay: 50 ms
Thread 6, 0    variSpeed delay: 94 ms
Thread 5, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 6, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 5, 0    resuming after variSpeed delay
Thread 5, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 5, 0    END thread
Thread 5, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 1
[0, 1, 3, 2, 5]
Thread 6, 0    resuming after variSpeed delay
Thread 6, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 6, 0    variSpeed delay: 84 ms
Thread 4, 0    variSpeed delay: 53 ms
Thread 6, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 4, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 4, 0    resuming after variSpeed delay
Thread 4, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 4, 0    END thread
Thread 4, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 0
[0, 1, 3, 2, 5, 4]
Thread 6, 0    resuming after variSpeed delay
Thread 6, 0    Sems: KICK 0 Q: 5; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 7, 0    variSpeed delay: 42 ms
Thread 7, 0    Sems: KICK 0 Q: 4; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 7, 0    resuming after variSpeed delay
Thread 7, 0    Sems: KICK 0 Q: 4; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 7, 0    variSpeed delay: 90 ms
Thread 7, 0    Sems: KICK 0 Q: 4; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 7, 0    resuming after variSpeed delay
Thread 7, 0    Sems: KICK 0 Q: 4; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 7, 0    variSpeed delay: 49 ms
Thread 7, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 8, 0    variSpeed delay: 23 ms
Thread 8, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 8, 0    resuming after variSpeed delay
Thread 8, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 6, 0    variSpeed delay: 46 ms
Thread 8, 0    variSpeed delay: 39 ms
Thread 6, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 8, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 7, 0    resuming after variSpeed delay
Thread 7, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 7, 0    END thread
Thread 7, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 0
[0, 1, 3, 2, 5, 4, 7]
Thread 8, 0    resuming after variSpeed delay
Thread 8, 0    Sems: KICK 0 Q: 3; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 9, 0    variSpeed delay: 32 ms
Thread 9, 0    Sems: KICK 0 Q: 2; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 6, 0    resuming after variSpeed delay
Thread 6, 0    Sems: KICK 0 Q: 2; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 6, 0    END thread
Thread 6, 0    Sems: KICK 0 Q: 2; ENTER 0 Q: 0; HOLD:0 Q: 1
```

```
[0, 1, 3, 2, 5, 4, 7, 6]
Thread 9, 0      resuming after variSpeed delay
Thread 9, 0      Sems: KICK 0 Q: 2; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 9, 0      variSpeed delay: 81 ms
Thread 9, 0      Sems: KICK 0 Q: 2; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 9, 0      resuming after variSpeed delay
Thread 9, 0      Sems: KICK 0 Q: 2; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 9, 0      variSpeed delay: 39 ms
Thread 9, 0      Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 10, 0     variSpeed delay: 99 ms
Thread 10, 0     Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 9, 0      resuming after variSpeed delay
Thread 9, 0      Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 9, 0     END thread
Thread 9, 0      Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 1
[0, 1, 3, 2, 5, 4, 7, 6, 9]
Thread 10, 0     resuming after variSpeed delay
Thread 10, 0     Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 8, 0      variSpeed delay: 69 ms
Thread 10, 0     variSpeed delay: 77 ms
Thread 8, 0      Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 10, 0     Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 8, 0      resuming after variSpeed delay
Thread 8, 0      Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 8, 0     END thread
Thread 8, 0      Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 0
[0, 1, 3, 2, 5, 4, 7, 6, 9, 8]
Thread 10, 0     resuming after variSpeed delay
Thread 10, 0     Sems: KICK 0 Q: 1; ENTER 0 Q: 0; HOLD:0 Q: 0
Thread 11, 0     variSpeed delay: 3 ms
Thread 11, 0     Sems: KICK 0 Q: 0; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 11, 0     resuming after variSpeed delay
Thread 11, 0     Sems: KICK 0 Q: 0; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 11, 0     variSpeed delay: 69 ms
Thread 11, 0     Sems: KICK 0 Q: 0; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 11, 0     resuming after variSpeed delay
Thread 11, 0     Sems: KICK 0 Q: 0; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 11, 0     variSpeed delay: 54 ms
Thread 11, 0     Sems: KICK 1 Q: 0; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 11, 0     resuming after variSpeed delay
Thread 11, 0     Sems: KICK 1 Q: 0; ENTER 0 Q: 0; HOLD:0 Q: 1
Thread 11, 0     END thread
Thread 11, 0     Sems: KICK 1 Q: 0; ENTER 0 Q: 0; HOLD:0 Q: 1
[0, 1, 3, 2, 5, 4, 7, 6, 9, 8, 11]
```