Distributed Computing

Lamport's Logical Clock

Yong Kim

What to do

~~1. Write clear pseudocode for each algorithm and submit it as a PDF report.~~

~~2. Implement your algorithm in C/C++~~

~~3. Compile and execute the program using the examples provided.~~

~~4. Create a file with the output of the program for an input value and submit it together with the program. Note, the output can be redirected to a file (for easy printing).~~

output will be saved in a file named

output.txt

How to Execute

```
1.  ~/474$ make                    // make an executable file, LC
2.  ~/474$ ./LC testfile.txt       // execute LC with argument file 'testfile.txt'
3.  ~/474$ cat output.txt          // check the output
```

Limitations

- If the first character of the file is white space(s), and the file needs to be verified, this file will not work well.

Algorithm Calculate

```
create a [ process x event ] matrix based on input.txt

set every processes basic logical time as 1 to # of events

// update logical time and check if receive events can execute

while ( every event is not valid ) {                    // 'valid' means, an event can execute
    for process in matrix {                             // e.g. receive event cannot be valid
        for event in each process {                     // if its send event is not valid
            if (event is internal) {
                logical time = previous logical time + 1
            }
            if (event is sending) {
                logical time = previous logical time + 1
                the associated recv event's logical time = max( logical time + 1, basic lc)
            }
            if (event is receiving) {
                if (this event is visited before) {
                    INCORRECT PROCESS
                    exit
                } else {
                    save this event
                }
                if (this event is valid) {
                    logical time = max(current logical time, previous logical time + 1)
                }
print logical times
```

Algorithm Verify

```
create a [ process x event ] matrix based on input.txt
// verify receive event
int i = 1
for ( int i = 0; i < # of process; i++ ) {
    for ( int j = 0; j < process size; j++ ) {
        if (logical time - previous logical time > 1) {
            current event is receiver[ k ]            // k is label for sender and receiver
            sender's logical time = logical time – 1
            i++
        } else if ( logical time – previous logical time == 1) {
            keep going
        } else {
            WRONG PROCESS        // current logical time – previous logical time <= 0
        }
// find and update send event
for (int k = 0; k < # of receiver; k++) {        // k is label
    for (int i = 0; i < # of process; i++) {
        for (int j = 0; j < process size; j++) {
            if ( current logical time == receiver's logical time – 1) {
                current event is sender[ k ]
            } else {
                continue;
            }
        }
    }
}
print events
```