

## 14.2

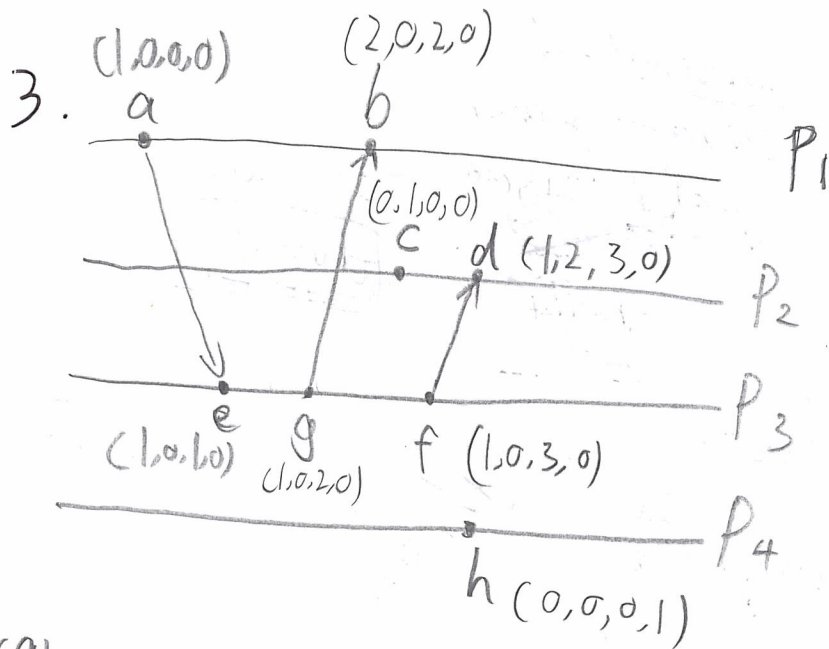
If to set it back to 10:27:50, it is possible for a source file to be compiled after 10:27:50 but before 10:27:54, the source file might appear to have been modified prior to the compilation. Erroneously, UNIX make facility will not recompile the source file. We only need to change the rate at which updates are made to the time as given to applications. This can be achieved in software without changing the rate at which the underlying hardware clock ticks.

## 14.4

It should use the third time, because the round-trip time is the minimum compared to two other times. Set the time to be  $10:54:28.342 + 0.01s = 10:54:28.352$  and the accuracy is  $\pm 10ms$ . If the time between sending and receiving is 8 ms, the accuracy improves to  $\pm 2ms$

## 14.14

1. Process P saves its state first, and then create a marker message and process p send this marker message to the process q, and then turn on the recording channel Cqp,
2. process q saves its state first, and then mark the state of channel Cqp as empty. And then process q starts recording the incoming messages on each of the incoming channels. And then start sending out the markers



(a)

(b)  $d(1,2,3,0)$

```

inconsistent(int[] v, int[] w) {
    bool greater = false, less = false
    for (int i = 0; i < v.length; i++) {
        if (v[i] > w[i]) { greater = true; }
        if (v[i] < w[i]) { less = true; }
    }
    if (greater && less) { return true; }
    else { return false; }
}

```

based on this function, event  $b$  and  $h$  are concurrent with event  $d$ .

4. (a)  $T_{round} = 6:22:15.250 - 6:22:15.100 = 0.15s$   
 client's clock:  $6:21:10.700 + 0.15s/2 = 6:21:10.775$

(b)  $124ms/2 = 62ms$