Real-Time Systems

Exercise 3: Time and Order

Global Time and Clock Drift

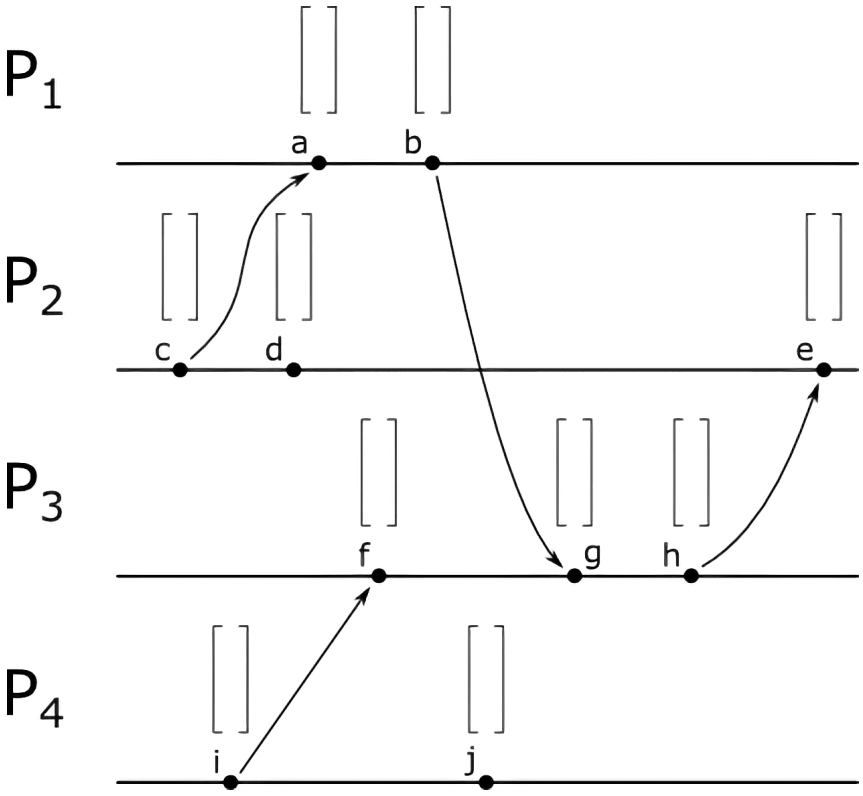
a) Describe a situation where drifting clocks in a distributed system may lead to problems.

b) Given a latency jitter of 20 µs, a clock drift rate of 10^{-5}, and a synchronization period of 1 second, what precision can be achieved by a central master algorithm?

Two types of events are received by two nodes comprising a system. Each node receives one event type and logs the receive time. From those logs, we want to later establish chronological order, whenever two events arrived at least 150µs apart.

c) Given a drift rate of 10^{-5} and a latency jitter of 20µs between both clocks, how often must both clocks be synchronized? What precision can be achieved that way?

Vector Clocks



a) Amend the diagram with the missing vector times. Use [0000] as initialization.

b) Which causal dependencies can be retrieved from the diagram?

c) Name two event pairs about which you can not retrieve dependency information and justify this with vector time properties.

d) Move events b, d, and j to the right as far as possible, without destroying causal dependencies. Substantiate your actions.

Programming with Clocks

We want to implement a periodic task with a period of 10ms. The operating system offers the following primitives:

- `sleep()` suspend execution until a relative time instant
- `sleep_until()` suspend execution until an absolute time instant
- `atomic { }` short code section where preemption is prevented

Explain the problems with the following program:

```
while (1) {
      work();  // takes one 1ms in average
      sleep(9ms);
}
```

Explain the problems with the following program:

```
while (1) {
      start = now();
      work();
      end = now();
      delay = 10ms - (end - start);
      sleep(delay);
}
```

Now propose solutions to these problems with the given operating system primitives.