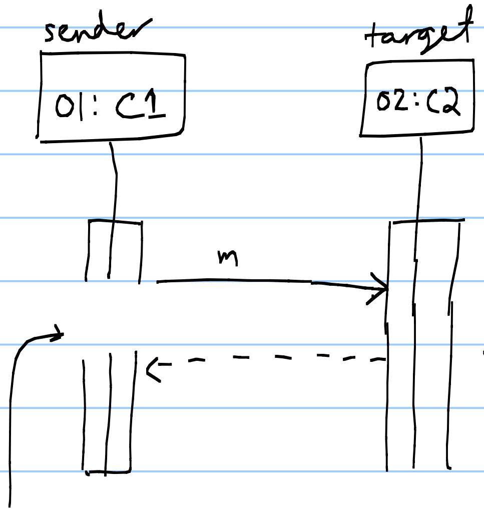
Comp 304 REEHAN SHAIKH

Note Title 3/3/2006

Synchronous msg:



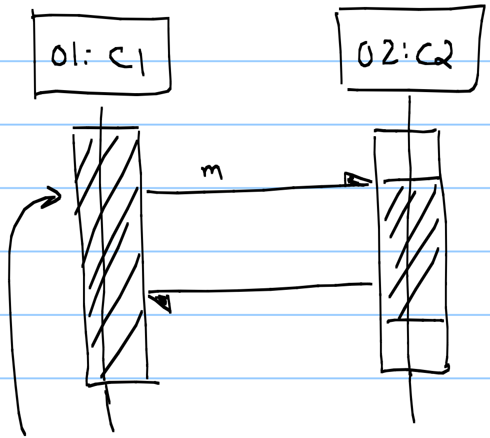
① sender wait (blocks) for response

② single thread of exec.

③ target process accepts a single call at a time so a 3rd class cannot send a msg to O2

② after sending msg, O1 blocks

Asynchronous msg:



① Sender DOES NOT wait for response

② Multiple thread of exec.

③ Target process may accept multiple call at same time

② sender doesn't block

③ If some class O3 calls O2, O2 can do many things.

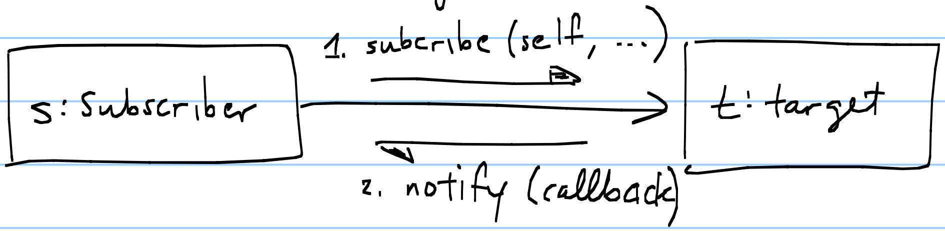
0. Ignore call
1. interrupt: start new call, leave old call.
2. spawn a new O2 thread and deal w/ new call.

process: own state/memory
threads: share memory/data (be careful)

3. queue calls (scheduling) (priority) (FIFO)
for priority, we can limit certain calls to have certain priorities, otherwise, every msg would have high priority, defeating the purpose of priority queues.

Callback

1) Subscriber object registers itself w/ a target.



ie: Observer pattern: subscribe to some interest, if interest is changed, target will send a "notify" to let subscriber know something changed w/ its interest.