MODULE *NJupiter*

```
********** Original Jupiter algorithm **********
class Msg
// type: [op, seq]

class Client
    var lseq     // local generated msg number, init to 0
    var gseq     // global received msg number, init to 0
    var outgoing // [op, seq]

    synchronized procedure Do(op):
        Apply(op)
        Send(the other side, [op, gseq])
        Append(outgoing, [op, lseq])
        lseq := lseq + 1

    synchronized procedure Recv(msg):
        RemoveIf(outgoing, lambda i : i.seq < msg.seq)
        xop, outgoing := Xform(msg.op, outgoing)
        Apply(xop)
        gseq := gseq + 1

class ServerThread// every client has a corresponding server thread
    var lseq// in fact, its meaning is gseq in server's scenario
            // for reusing Client funcs, I don't exchange their usages
    var gseq
    var outgoing

    synchronized procedure SRecv(msg):
        Client.Recv(msg)// just reuse the code
        SignalOtherServerThreads(xop)

    synchronized procedure Signaled(op):
        Client.Do(op)     // just reuse the code
```

EXTENDS *JupiterInterface*, *OT*, *BufferStateSpace*

VARIABLES

cbuf, *cseq*, *sbuf*, *sseq*

cbuf[*c*]: client outgoing: [*op*, *seq*]

cseq[*c*]: client lseq gseq: [*l*, *g*]

$vars \triangleq \langle intVars, cbuf, cseq, sbuf, sseq \rangle$

$NMsg \triangleq$

$[c : Client,\ seq : Nat,\ op : Op \cup \{Nop\}] \cup$ client $\rightarrow$ server
$[seq : Nat,\ op : Op \cup \{Nop\}]$ server $\rightarrow$ client

---

$TypeOK \triangleq$
$\quad \land \quad TypeOKInt$
$\quad \land \quad cbuf \in [Client \rightarrow Seq([op : Op \cup \{Nop\},\ seq : Nat])]$
$\quad \land \quad cseq \in [Client \rightarrow [l : Nat,\ g : Nat]]$
$\quad \land \quad cbuf \in [Client \rightarrow Seq([op : Op \cup \{Nop\},\ seq : Nat])]$
$\quad \land \quad cseq \in [Client \rightarrow [l : Nat,\ g : Nat]]$

$Init \triangleq$
$\quad \land\ InitInt$
$\quad \land\ cbuf = [c \in Client \mapsto \langle\rangle]$
$\quad \land\ cseq = [c \in Client \mapsto [l \mapsto 0,\ g \mapsto 0]]$
$\quad \land\ sbuf = [c \in Client \mapsto \langle\rangle]$
$\quad \land\ sseq = [c \in Client \mapsto [l \mapsto 0,\ g \mapsto 0]]$

$RemoveAckedOps(s,\ seq) \triangleq$
$\quad$ LET $F[i \in 0\ ..\ Len(s)] \triangleq$
$\qquad$ IF $i = 0$
$\qquad$ THEN $\langle\rangle$
$\qquad$ ELSE IF $s[i].seq \geq seq$
$\qquad\qquad$ THEN $Append(F[i-1],\ s[i])$
$\qquad\qquad$ ELSE $F[i-1]$
$\quad$ IN $F[Len(s)]$

$OTWrapper(l,\ r) \triangleq [op \mapsto OT(l.op,\ r.op),\ seq \mapsto l.seq]$

$ClientPerform(c,\ m) \triangleq$
$\quad$ LET $xform \triangleq xFormFull(OTWrapper,\ m,\ RemoveAckedOps(cbuf[c],\ m.seq))$
$\quad$ IN $\land\ cbuf' = [cbuf$ EXCEPT $![c] = xform.xops]$
$\qquad \land\ cseq' = [cseq$ EXCEPT $![c] = [l \mapsto @.l,\ g \mapsto @.g + 1]]$
$\qquad \land\ SetNewAop(c,\ xform.xop.op)$

$ServerPerform(m) \triangleq$
$\quad$ LET $c \triangleq m.c$
$\qquad xform \triangleq xFormFull(OTWrapper,\ m,\ RemoveAckedOps(sbuf[c],\ m.seq))$
$\qquad xop \triangleq xform.xop.op$
$\quad$ IN $\land\ sseq' = [cl \in Client \mapsto$ IF $cl = c$
$\qquad\qquad\qquad\qquad\qquad\qquad$ THEN $[l \mapsto sseq[cl].l + 1,\ g \mapsto sseq[cl].g]$
$\qquad\qquad\qquad\qquad\qquad\qquad$ ELSE $[l \mapsto sseq[cl].l,\ g \mapsto sseq[cl].g + 1]]$
$\qquad \land\ sbuf' = [cl \in Client \mapsto$ IF $cl = c$
$\qquad\qquad\qquad\qquad\qquad\qquad$ THEN $xform.xops$
$\qquad\qquad\qquad\qquad\qquad\qquad$ ELSE $Append(sbuf[cl],\ [op \mapsto xop,\ seq \mapsto sseq[cl].g])]$
$\qquad \land\ SetNewAop(Server,\ xop)$
$\qquad \land\ Comm!SSend(c,\ [cl \in Client \mapsto [seq \mapsto sseq[cl].l,\ op \mapsto xop]])$

$DoOp(c,\ op) \triangleq$
$\quad \land\ SetNewAop(c,\ op)$
$\quad \land\ cbuf' = [cbuf\ \text{EXCEPT}\ ![c] = Append(@,\ [op \mapsto op,\ seq \mapsto cseq[c].l])]$
$\quad \land\ cseq' = [cseq\ \text{EXCEPT}\ ![c] = [l \mapsto @.l + 1,\ g \mapsto @.g]]$
$\quad \land\ Comm!CSend([c \mapsto c,\ seq \mapsto cseq[c].g,\ op \mapsto op])$

$Do(c) \triangleq$
$\quad \land\ DoInt(DoOp,\ c)$
$\quad \land\ \text{UNCHANGED}\ \langle sbuf,\ sseq\rangle$

$Rev(c) \triangleq$
$\quad \land\ RevInt(ClientPerform,\ c)$
$\quad \land\ \text{UNCHANGED}\ \langle sbuf,\ sseq\rangle$

$SRev \triangleq$
$\quad \land\ SRevInt(ServerPerform)$
$\quad \land\ \text{UNCHANGED}\ \langle cbuf,\ cseq\rangle$

$Next \triangleq$
$\quad \lor \exists\, c \in Client : Do(c) \lor Rev(c)$
$\quad \lor\ SRev$

$Fairness \triangleq$
$\quad \text{WF}_{vars}(SRev \lor \exists\, c \in Client : Rev(c))$

$Spec \triangleq\ Init \land \Box[Next]_{vars}$ $\land\ Fairness$

$QC \triangleq$ Quiescent Consistency
$\quad Comm!EmptyChannel \Rightarrow Cardinality(Range(state)) = 1$

THEOREM $Spec \Rightarrow \Box QC$

---

* Modification History
* Last modified Sun *Apr* 21 09:37:06 *CST* 2019 by *tangruize*
* Last modified *Thu Jan* 17 10:30:39 *CST* 2019 by *hengxin*
* Created *Satchins*, Jun 23 17:14:18 *CST* 2018 by *hengxin*