```
┌──────────────────── MODULE solution ────────────────────┐
EXTENDS Naturals, Sequences

CONSTANTS
    MAX_TEXT_LINES,
    MAX_USERS,
    MAX_LOG_SIZE

VARIABLES
    file,
    log

TEXT_LINES ≜ 1 . . MAX_TEXT_LINES
INIT_VALUE ≜ 1
SERVER ≜ 0
USERS ≜ 1 . . MAX_USERS
PARTICIPANTS ≜ {SERVER} ∪ USERS
PRIMES ≜
    ⟨2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107⟩

UserId(u) ≜
    PRIMES[u]

LineId(i) ≜
    PRIMES[MAX_USERS + 1 + i]

NewFile(user, line_num) ≜
    [file EXCEPT ![user][line_num] = @ ∗ UserId(user) ∗ LineId(line_num)]

LastLogEntry(user) ≜
    log[user][Len(log[user])]

Modified(user) ≜
    LastLogEntry(user).file ≠ file[user]

Write(user, line_num) ≜
    ∧ user ∈ USERS
    ∧ line_num ∈ TEXT_LINES
    ∧ Modified(user) = FALSE
    ∧ file′ = NewFile(user, line_num)
    ∧ UNCHANGED log

NewLogEntry(user) ≜
    [user ↦ user, file ↦ file[user]]

Commit(user) ≜
    ∧ Modified(user) = TRUE
    ∧ Len(log[user]) < MAX_LOG_SIZE
    ∧ log′ = [log EXCEPT ![user] = Append(@, NewLogEntry(user))]
    ∧ UNCHANGED file

IsPrefixOf(seq1, seq2) ≜
    ∧ Len(seq1) ≤ Len(seq2)
    ∧ seq1 = SubSeq(seq2, 1, Len(seq1))

AbleToPush(user) ≜
    ∧ log[SERVER] ≠ log[user]
    ∧ IsPrefixOf(log[SERVER], log[user])

Push(user) ≜
     ∧ AbleToPush(user)
     ∧ log′ = [log EXCEPT ![SERVER] = log[user]]
     ∧ file′ = [file EXCEPT ![SERVER] = LastLogEntry(user).file]

AbleToPull(user) ≜
    ∧ log[SERVER] ≠ log[user]
    ∧ IsPrefixOf(log[user], log[SERVER])

Pull(user) ≜
     ∧ AbleToPull(user)
     ∧ log′ = [log EXCEPT ![user] = log[SERVER]]
     ∧ file′ = [file EXCEPT ![user] = file[SERVER]]

DivergentIndex(user) ≜
    LET
        server_log ≜ log[SERVER]
        user_log ≜ log[user]
        Min(a, b) ≜ IF a < b THEN a ELSE b
        min_length ≜ Min(Len(server_log), Len(user_log))
    IN
        IF IsPrefixOf(server_log, user_log) ∧ (Len(server_log) < Len(user_log))
         THEN Len(server_log) + 1
         ELSE CHOOSE i ∈ 1 . . min_length :
            ∧ server_log[i].file ≠ user_log[i].file
            ∧ ∀ j ∈ 1 . . (i − 1) : server_log[j].file = user_log[j].file

ChangedLineAndValue(file1, file2) ≜
    LET ci ≜ CHOOSE i ∈ TEXT_LINES : file1[i] ≠ file2[i]IN
         [line_num ↦ ci, value ↦ file2[ci]]

ChangedEntries(user) ≜
    IF log[SERVER] = log[user]
     THEN ⟨⟩
     ELSE LET
        start ≜ DivergentIndex(user)
    $FileOf(i) \triangleq log[user][i + start - 1].file$
  IN
    $[i \in 1 \,..\, (Len(log[user]) - start + 1) \mapsto ChangedLineAndValue(FileOf(i - 1), FileOf(i))]$

RECURSIVE $ApplyDivergence(\_, \_, \_)$
$ApplyDivergence(user, new\_log, changed\_entries) \triangleq$
  LET
    $ChangeFile(target, changed\_entry) \triangleq$
      $[target$ EXCEPT $![changed\_entry.line\_num] = changed\_entry.value]$
    $UpdateLog(old\_log, changer, changed\_entry) \triangleq$
      LET $log\_record \triangleq [$
          $user \mapsto changer,$
          $file \ \ \mapsto ChangeFile($
            $old\_log[Len(old\_log)].file,$
            $changed\_entry$
          $)$
      $]$IN  $Append(old\_log, log\_record)$
  IN
    IF $changed\_entries = \langle\rangle$
     THEN $new\_log$
     ELSE $ApplyDivergence(user, UpdateLog(new\_log, user, Head(changed\_entries)), Tail(changed\_entrie$

$AbleToMergeButNotPullOrPush(user) \triangleq$
   $\land AbleToPull(user)$  $=$ FALSE
   $\land AbleToPush(user) =$ FALSE
   $\land \exists\, i \in$ DOMAIN $log[SERVER]$ :
       $\land i \in$ DOMAIN $log[user]$
       $\land log[user][i] \neq log[SERVER][i]$

$Merge(user) \triangleq$
   $\land user \in USERS$
   $\land AbleToMergeButNotPullOrPush(user)$
   $\land log' = [$
     $log$ EXCEPT $![user] = ApplyDivergence($
       $user,$
       $log[SERVER],$
       $ChangedEntries(user))]$
   $\land file' = [file$ EXCEPT $![user] = log'[user][Len(log'[user])].file]$

$WriteAction \triangleq$
  $\exists\, user \in USERS, line\_num \in TEXT\_LINES$ :
    $\land Write(user, line\_num)$

$CommitAction \triangleq$
  $\exists\, user \in USERS$ :
    $Commit(user)$
```

$$\begin{array}{l} \mathit{PushAction} \triangleq \\ \quad \exists\, \mathit{user} \in \mathit{USERS} : \\ \qquad \mathit{Push}(\mathit{user}) \end{array}$$

$$\begin{array}{l} \mathit{PullAction} \triangleq \\ \quad \exists\, \mathit{user} \in \mathit{USERS} : \\ \qquad \mathit{Pull}(\mathit{user}) \end{array}$$

$$\begin{array}{l} \mathit{MergeAction} \triangleq \\ \quad \exists\, \mathit{user} \in \mathit{USERS} : \\ \qquad \mathit{Merge}(\mathit{user}) \end{array}$$

$$\begin{array}{l} \mathit{InitFile} \triangleq [\mathit{new\_log} \in \mathit{TEXT\_LINES} \mapsto \mathit{INIT\_VALUE}] \\ \mathit{InitLog} \triangleq \langle [\mathit{user} \mapsto \mathit{SERVER}, \mathit{file} \quad\quad \mapsto \mathit{InitFile}] \rangle \end{array}$$

$$\begin{array}{l} \mathit{Init} \triangleq \\ \quad \land\ \mathit{file} = [u \in \mathit{PARTICIPANTS} \mapsto \mathit{InitFile}] \\ \quad \land\ \mathit{log} = [p \in \mathit{PARTICIPANTS} \mapsto \mathit{InitLog}] \end{array}$$

$$\begin{array}{l} \mathit{Next} \triangleq \\ \quad \lor\ \mathit{WriteAction} \\ \quad \lor\ \mathit{CommitAction} \\ \quad \lor\ \mathit{PushAction} \\ \quad \lor\ \mathit{PullAction} \\ \quad \lor\ \mathit{MergeAction} \end{array}$$

$$\begin{array}{l} \mathit{vars} \triangleq \langle \mathit{file}, \mathit{log} \rangle \\ \mathit{Spec} \triangleq \\ \quad \land\ \mathit{Init} \\ \quad \land\ \Box[\mathit{Next}]_{\mathit{vars}} \\ \quad \land\ \mathrm{WF}_{\mathit{vars}}(\mathit{Next}) \end{array}$$

$$\begin{array}{l} \mathit{FileType} \triangleq [\mathit{TEXT\_LINES} \rightarrow \mathit{Nat}] \\ \mathit{LogEntry} \triangleq [\mathit{user} : \mathit{PARTICIPANTS}, \mathit{file} : \mathit{FileType}] \end{array}$$

$$\begin{array}{l} \mathit{TypeOK} \triangleq \\ \quad \land\quad \mathit{file} \in [\mathit{PARTICIPANTS} \rightarrow \mathit{FileType}] \\ \quad \land\quad \textsc{domain}\ \mathit{log} = \mathit{PARTICIPANTS} \\ \quad \land\quad \forall\, u \in \mathit{PARTICIPANTS} : \forall\, i \in \textsc{domain}\ \mathit{log}[u] : i \geq 1 \\ \quad \land\quad \forall\, u \in \mathit{PARTICIPANTS} : \forall\, i \in \textsc{domain}\ \mathit{log}[u] : \mathit{log}[u][i] \in \mathit{LogEntry} \end{array}$$

$$\mathit{GLOBAL\_MAX\_LOG\_SIZE} \triangleq (\mathit{MAX\_LOG\_SIZE} - 1) * \mathit{MAX\_USERS} + 1$$

$$\begin{array}{l} \mathit{LogIsBounded} \triangleq \\ \quad \lor\ \forall\, u \in \mathit{PARTICIPANTS} : \mathit{Len}(\mathit{log}[u]) \leq \mathit{GLOBAL\_MAX\_LOG\_SIZE} \end{array}$$

$$\begin{array}{l}
\mathit{LastLogEntryIsFile} \triangleq \\
\quad \forall\, u \in \mathit{PARTICIPANTS} : \mathit{Modified}(u) = \textsc{false} \Rightarrow \mathit{LastLogEntry}(u).\mathit{file} = \mathit{file}[u]
\end{array}$$

$$\begin{array}{l}
\mathit{WriteOnlyOnce} \triangleq \\
\quad \forall\, u \in \mathit{USERS} : \exists\, \mathit{line\_num} \in \mathit{TEXT\_LINES} : \\
\qquad \textsc{enabled}\ \mathit{Write}(u,\, \mathit{line\_num}) \Rightarrow \neg(\textsc{enabled}\ \mathit{Commit}(u))
\end{array}$$

$$\begin{array}{l}
\mathit{LogAlwaysGrows} \triangleq \\
\quad \exists\, n \in \mathit{MAX\_LOG\_SIZE} \,..\, \mathit{GLOBAL\_MAX\_LOG\_SIZE} : \\
\qquad \forall\, u \in \mathit{PARTICIPANTS} : \Diamond(\mathit{Len}(\mathit{log}[u]) = n)
\end{array}$$

Consequent entries should differ by only one line

$$\begin{array}{l}
\mathit{DiffBetweenConsequentEntries} \triangleq \\
\quad \forall\, u \in \mathit{USERS} : \forall\, i \in \textsc{domain}\ \mathit{log}[u] : i + 1 \in \textsc{domain}\ \mathit{log}[u] \Rightarrow \\
\qquad \exists\, \mathit{line\_num} \in \mathit{TEXT\_LINES} : \\
\qquad\quad \land\ \mathit{log}[u][i].\mathit{file}[\mathit{line\_num}] \neq \mathit{log}[u][i+1].\mathit{file}[\mathit{line\_num}] \\
\qquad\quad \land\ \forall\, \mathit{other\_line\_num} \in \mathit{TEXT\_LINES} : \mathit{other\_line\_num} \neq \mathit{line\_num} \Rightarrow \\
\qquad\qquad\quad \mathit{log}[u][i].\mathit{file}[\mathit{other\_line\_num}] = \mathit{log}[u][i+1].\mathit{file}[\mathit{other\_line\_num}]
\end{array}$$

$$\begin{array}{l}
\mathit{LogsConverge} \triangleq \\
\quad \neg(\textsc{enabled}\ \mathit{Next}) \Rightarrow \\
\qquad (\forall\, u1 \in \mathit{PARTICIPANTS},\ u2 \in \mathit{PARTICIPANTS} : \mathit{log}[u1] = \mathit{log}[u2])
\end{array}$$

$$\begin{array}{l}
\mathit{PullOrPush} \triangleq \\
\quad \land\ \forall\, u \in \mathit{USERS} : \textsc{enabled}\ \mathit{Pull}(u) \Rightarrow \\
\qquad\quad \neg(\textsc{enabled}\ \mathit{Push}(u)) \\
\quad \land\ \forall\, u \in \mathit{USERS} : \textsc{enabled}\ \mathit{Push}(u) \Rightarrow \\
\qquad\quad \neg(\textsc{enabled}\ \mathit{Pull}(u))
\end{array}$$

$$\begin{array}{l}
\mathit{NoDirectChangesOnServer} \triangleq \\
\quad \land\ \neg\mathit{Modified}(\mathit{SERVER})
\end{array}$$

$$\begin{array}{l}
\mathit{EventuallyUnableToCommit} \triangleq \\
\quad \forall\, u \in \mathit{USERS} : \Diamond\Box\neg(\textsc{enabled}\ \mathit{Commit}(u))
\end{array}$$

$$\begin{array}{l}
\mathit{LogReachesMaxLength} \triangleq \\
\quad \exists\, n \in \mathit{MAX\_LOG\_SIZE} \,..\, \mathit{GLOBAL\_MAX\_LOG\_SIZE} : \\
\qquad \forall\, u \in \mathit{USERS} : \Diamond\Box(\mathit{Len}(\mathit{log}[u]) = n)
\end{array}$$