

CLAIMS MANAGER

Paul Glassco

July 5, 1974

Systems Group

Technical Document

Center for Research in Management Science

University of California

Berkeley

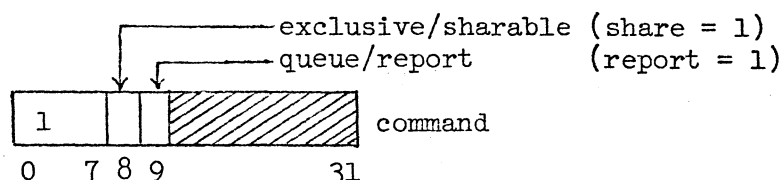
This work was done as part of the Systems Development effort under National Science Foundation Grant GS-32138.

The Claims Manager is a process which may voluntarily be used by user processes to coordinate accesses to shared objects. A user process may request an exclusive or a sharable claim for a specified object by sending the appropriate IPC request to the Claims Manager. When a user process is granted a successful exclusive claim to an object, then no other user process will be granted a claim for the object. When a user process is granted a sharable claim to an object, then other user processes which also request sharable access to the object (and which specify identical access options to the object) will be granted a simultaneous claim. A user process may UNCLAIM an object previously claimed. When a user process is destroyed, the Job Manager automatically executes a call on the Claim Manager requesting that all claims made by the designated user process be unclaimed.

The three requests to the claims manager are transmitted via the IPC message mechanism. An outline of the three requests follows.

CLAIM

Arguments: a scalar command of 1 (in the first byte) with two option bits, a port capability for response and an object capability.

Format:Action:

First, a check is made of the number of requests which the given process currently has. If there are already as many as the limit, the present request is denied. Otherwise, action continues in two possible ways, depending upon whether the request is to be queued or reported back if not immediately obtainable. We examine these separately.

QUEUE: A request is registered for the given process. If the request is exclusive and the object is immediately available, the request is granted and a message of scalar value 0 is returned to the process. If the request is exclusive and the object is not immediately obtainable, the request is queued at the end of the request queue for that object. If the request is sharable and the (currently) last request in the request queue (and hence everyone in the request queue) is sharing the object and the options are identical, the request is granted and a message of scalar value 0 is returned to the process. If the request is sharable but sharing is not compatible with other requests in the request queue, the current request is queued at the end. Requests which are not immediately granted are serviced on a first-come-first-served basis.

Assuming that other requests are served in finite time, a queued request will eventually be granted a claim on the object, at which time a message with scalar value 0 is returned to the requesting process. In all cases a message is ultimately returned.

REPORT: If the request is exclusive and the object is immediately available, the request is granted and a message of scalar value 0 is returned to the process. If the request is exclusive and the object is not immediately available, the request is denied and a message of scalar value 5 is returned. If the request is sharable and sharing is compatible, the request is granted and a message of scalar value 0 is returned. If the request is sharable and sharing is not compatible, the request is denied and a message of scalar value 5 is returned. In all cases, a message is immediately returned.

Errors

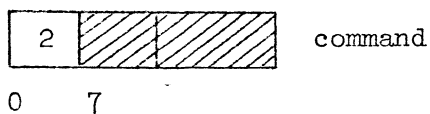
Any return message with scalar value $\neq 0$ indicates failure to grant the request as presented. The return message is sent only if a valid port capability is sent with the request. If no command is specified, a message with the scalar value 1 is returned. If the command sent is not a 1, 2, or 3, a message of scalar value 2 is returned. If no object capability is given a message with scalar value 3 is returned. If the process already has the maximum number of requests, a scalar value of 4 is returned. Note that this count is of the number of claims requested, not necessarily the number of claims granted. If the process wished to be reported rather than to be queued, and the object was not available, a message with scalar value 5 is returned. If the claims manager can support no more requests, a message with scalar value of 6 is sent back.

<u>Returned Value</u>	<u>Meaning</u>
1	no command sent
2	command not 1, 2, or 3
3	no object capability given
4	process already has maximum number of requests
5	object not immediately available and process wishes to be told about it
6	claims manager saturated

UNCLAIM

Arguments: A command of 2 and an object capability.

Format:



Action:

If no object capability is given, nothing is done. If a claim by the process for the specified object is found, that request is cleared. (If there is more than one then the oldest request is deleted.) If no claim request is found, nothing is done.

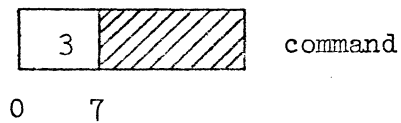
Errors:

There are no error messages returned.

PURGE\ALL\CLAIMS

Arguments: A command of 3 and a process capability with "purge claim" option bit.

Format:



Action:

All claims by the given process are deleted.

Errors:

There are no error messages returned.