

PUBLISHED: 6/7/67
Minor Revision
(Supercedes BG.6.03,
6/6/66)

Identification

The replenisher
P. G. Neumann and M. R. Wagner

Purpose

The replenisher attempts to keep the size of the free pool at a level high enough to satisfy system requirements and low enough to avoid waste. It does so by removing hyperpages from core on the basis of lack of use.

Introduction

The replenisher is responsible for trying to maintain some sort of minimax strategy with respect to the supply of core in the free pool. In particular, it attempts to keep the free pool sufficiently large so that there is always an adequate supply of free core to satisfy the most extreme system urgencies. On the other hand, it also attempts to keep the supply of free core sufficiently small in order to reduce the amount of unnecessary removal of hyperpages.

The strategy which the replenisher adopts initially is always to write out enough pages to restore the potential supply of free core up to its roof. More sophisticated strategies may follow as required.

The desirability algorithm

The desirability of replenishing at a given time is based on the number nfree of blocks actually in the free pool, the number nevac of blocks which are currently evacuated, and the minimum number nfloor of blocks generally desired to be free or evacuated at any time. If nfree + nevac is smaller than nfloor, then it is desirable to replenish. Otherwise, it is not desirable to replenish. Once replenishing has been begun, it is continued until nfree + nevac is greater than or equal to nroof, the maximum number of blocks desired to be free. This double-limit strategy is employed to give the algorithm added flexibility.

Replenishing

If replenishing is desirable, the strategy is as follows. The group at the beginning of the eligible pool corresponding to pool in the assign call is removed from the list, and page control is called to remove the contents of that group. This group is by construction a reassignable group containing a hyperpage. If the return specifies immediate unassignment, the group type becomes free. If the return specifies eventual unassignment, the group

type becomes evacuated. If the return specifies unassignment is not currently possible, the group is returned to the end of the eligible list from whence it cometh. The next group from the beginning of the list is then chosen, and page control called. Sufficiently many groups from the beginning of the list are selected until it is no longer desirable to continue replenishing.