

Sequence of Page References	Window Size, $\Delta$			
	2	3	4	5
24	24	24	24	24
15	24 15	24 15	24 15	24 15
18	15 18	24 15 18	24 15 18	24 15 18
23	18 23	15 18 23	24 15 18 23	24 15 18 23
24	23 24	18 23 24	•	•
17	24 17	23 24 17	18 23 24 17	15 18 23 24 17
18	17 18	24 17 18	•	18 23 24 17
24	18 24	•	24 17 18	•
18	•	18 24	•	24 17 18
17	18 17	24 18 17	•	•
17	17	18 17	•	•
15	17 15	17 15	18 17 15	24 18 17 15
24	15 24	17 15 24	17 15 24	•
17	24 17	•	•	17 15 24
24	•	24 17	•	•
18	24 18	17 24 18	17 24 18	15 17 24 18

**Figure 8.19 Working Set of Process as Defined by Window Size**

Initial value of  $D_i$  is 0

After an operation, the value of  $D_i$  is updated as follows

- (I) if the next operation is a block allocate request:  
if there is any free block, select one to allocate  
if the selected block is locally free  
then  $D_i := D_i + 2$   
else  $D_i := D_i + 1$   
otherwise  
first get two blocks by splitting a larger one into two (recursive operation)  
allocate one and mark the other locally free  
 $D_i$  remains unchanged (but  $D$  may change for other block sizes because of the recursive call)
- (II) if the next operation is a block free request  
Case  $D_i = 2$   
mark it locally free and free it locally  
 $D_i := D_i - 2$   
Case  $D_i = 1$   
mark it globally free and free it globally; coalesce if possible  
 $D_i := 0$   
Case  $D_i = 0$   
mark it globally free and free it globally; coalesce if possible  
select one locally free block of size  $2i$  and free it globally; coalesce if possible  
 $D_i := 0$

**Figure 8.24 Lazy Buddy System Algorithm**