Frame Allocation Algorithms

Aleksander Wawrzyniak
April, the 10th 2019

An important aspect of operating systems - virtual memory is implemented using demand paging. Demand paging necessitates the development of a page replacement algorithm and a frame allocation algorithm. Frame Allocation Algorithms are used when we have multiple processes; it helps decide how many frames to allocate to each process.

There are various constraints to the strategies for the allocation of frames:

- **You cannot allocate more than the total number of available frames.**
- **At least a minimum number of frames should be allocated to each process.** This constraint is supported by two reasons. The first reason is, as less number of frames are allocated, there is an increase in the page fault ratio, decreasing the performance of the execution of the process. Secondly, there should be enough frames to hold all the different pages that any single instruction can reference.

Frame Allocation Algorithms

Equal allocation algorithm (static)

In a system with x frames and y processes, each process gets equal number of frames, i.e. x/y. We simply allocate floor(x/y) frames to every process we have.

Advantages of Equal algorithm

- Low resource
- Easy to implement

Disadvantages of Equal algorithm

- In systems with processes of varying sizes, it does not make much sense to give each process equal frames.
- Allocation of a large number of frames to a small process will eventually lead to the wastage of a large number of allocated unused frames.

Proportional allocation algorithms (static)

Frames are allocated to each process according to the process size. For a process p of size s, the number of allocated frames is `a = (s/S)*m`, where S is the sum of the sizes of all the processes and m is the number of frames in the system.

Advantages of Proportional algorithm

- Low resource
- All the processes share the available frames according to their needs, rather than equally.

Disadvantages of Proportional algorithm

- Still, if the process is trashing, no changes can be done.

Random allocation algorithm (static)

Random algorithm allocates random number of frames to each of the given processes. It is hard to predict how will it perform. It might be close to optimal, but also can be the worst one.

Advantages of Random algorithm

- Low resource
- Relatively easy to implement
- May be efficient

Disadvantages of Random algorithm

- May be very inefficient.
- Hard to predict.

Page Fault Frequency algorithm (dynamic)

Page fault frequency algorithm changes the number of pages allocated to processes until all processes are being served. It uses previously specified *interval* to decide when to check the page fault rate of the process and decide, whether it can take back frame(s) or allocate one more frame to it *(if there are any spare frames available at the moment)*. In my implementation, the algorithm takes the page when the page fault rate during the last interval was smaller than *30%* and allocates additional frame to the process when its page fault rate during the interval was higher than *80%*. When service of the process is done, all frames are taken back from it.

Advantages of Page Fault Frequency algorithm

- It should make page fault rate of the processes smaller, however it is not said it always will.

Disadvantages of Page Fault Frequency algorithm

- It is harder to implement
- It requires more computing resources than static algorithms.

Working Set algorithm

- The working set for each model is the set of pages referenced by the process during the most recent w page references.
- Dynamic adjustment of w is based on available memory, the number of processes and the page fault rate.
- Based on the available memory and number of processes, we can establish upper and lower bounds on the desired page fault rate. If the actual page fault rate for the process exceeds the upper limit, then we can increase w for the process. If the page fault rate falls below the lower limit, we decrease w.

- The number of frames allocated to the process at anytime is the number of unique page references over the last w references.

Advantages of Working Set algorithm

- It has lowest page fault rate among the given algorithms

Disadvantages of Working Set algorithm

- Hard to implement.
- Requires the most computing resources.

What algorithms are being used by the biggest operating systems

- Windows: **Working Set algorithm**
- Linux: **Working Set Algorithm**

Results (Page misses [%])

30 processes, 210 frames, interval: 12:

	Equal	Proportional	Random	PFF	WSA
1	33	38	80	64	40
2	35	39	83	62	40

35 processes, 250 frames, interval 14:

	Equal	Proportional	Random	PFF	WSA
1	34	37	73	59	40
2	35	38	86	60	41

Conclusions

From the results, we can see, that using the randomly generated pages in x processes (being numbers from 0 to 9) we can observe that:

- Equal algorithm has the best page fault rate, but it will be different for different sets of processes.
- The worst performance has random algorithm.
- Gathered values are highly dependent on randomly generated input sets. Therefore the two shown results may not be accurate enough, and will be different every time.

Disclaimer

Gathered results may not be highly accurate and should not be used as base for any scientific researches.