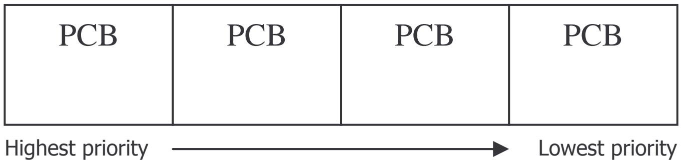
Prioritized Scheduling

- Priority byte(s) is/are added to the PCB. It is not important to the theory how many bytes are implemented.
- The ready queue becomes a priority queue, sorted by the priority byte.
- The priority queue is a sorted data structure.
- Challenge: preventing starvation
    - Technique: aging priority
        - Every X time unit, all processes that are not selected have their processes increased.

          $$At\ some\ \Delta t,\ \sum_{k=1}^{n-1} P[k] = P[k] + \Delta P$$

        - After a process ages to execution, the priority is reset to the original. This is a step many people forget to do.

Implementing a Priority Queue



Bubble Sort

Consider the set of numbers: 3, 12, 1, 9, 5
3, 1, 9, 5, 12
1, 3, 5, 9, 12

In C:

```
void BubbleSort( int* A, int n )
{
        int i;
        int temp;
        bool Done;

        do
        {

                Done = true;

                for (i = 0; i < n - 2; i++)
                {

                        if (A[i] > A[i+1]) {

                                temp = A[i];
                                A[i] = A[i+1];
                                A[i+1] = temp;

                                Done = false;

                        }

                }

        } while (!Done);
}
```

Selection Sort

```
void Selection (int *A, int n)
{

        int i, j, k, temp;

        i = n - 1;

        while (i > 0) {

                j = i;

                for (k = 0; k < i; k++) {

                        if (A[k] > A[j]) {

                                j = k;

                        }

                }

                temp = A[i];
                A[i] = A[j];
                A[j] = temp;

                i--;

        }

}
```