**Windows:**

* **Jobs:** É um objeto que permite que grupos de processos sejam gerenciados como uma unidade. A criação do Job é feita pela função CreateJobObject. Quando o Job é criado, ele não está associado a nenhum processo. Essa associação é feita pela função AssignProccessToJobObject, criando uma conexão que não pode ser quebrada. Se um processo está associado a um Job, qualquer processo filho dele também será associado ao Job.
* **Thread:** É uma coleção de Worker Threads que executam de forma eficiente retornos de chamadas assíncronas em nome da aplicação. A criação de uma Thread para um processo é feita pela função CreateThread. Ele possui 3 variáveis que criam todas as Worker Threads:
  + pDataArray[MAX\_THREADS]: Vetor de dados do tamanho máximo possível de Threads
  + dwThreadIdArray[MAX\_THREADS]: Vetor com id das Threads
  + hThreadArray[MAX\_THREADS]: Vetor com o header das Threads

Então, ele aloca memória para o dado da Thread, se a alocação falhar, o sistema está sem memória e apenas termina a execução. Após isso, gera um dado único para cada Thread trabalhar com ele, em seguida, cria-se a Thread para começar a execução. Após isso, verifica se a execução foi um sucesso, se falhar ela termina a execução. Por fim, ele chama a função WaitForMultipleObjects, que espera que todas as Threads tenham terminado para liberar memória.

* **Processo:** É um programa em execução. A criação de um Processo é feito pela função CreateProcess. O Processo possui uma variável de começo da informação (chamada de si) e uma variável com a informação do processo (chamada de pi). Então ele verifica se o Processo não possui memória alocada com a função ZeroMemory, e então ele aloca memória para si. Então, ele espera que um Processo seja criado com a função WaitForSingleObject para que ele possa alocar no espaço de si.