An informal scan of the real-time (embedded, dedicated, safety-critical) market:
* 30% assembly and legacy languages
* 30% Ada
* 30% C/C++
* 10% other (100+ other languages)

* C and Ada are the most commonly used languages in civil aviation today

* C++ is gaining popularity, but its usage is still


* Real-time languages
<table>
<thead>
<tr>
<th>Feature</th>
<th>RT-ADA</th>
<th>C/C++</th>
<th>JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory management</td>
<td>Automatic</td>
<td>Manual</td>
<td>Garbage collection</td>
</tr>
<tr>
<td>Run-time Efficiency</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Run-time Predictability</td>
<td>High</td>
<td>OS-dependent</td>
<td>Low</td>
</tr>
<tr>
<td>Concurrency controls</td>
<td>Language features</td>
<td>OS specific</td>
<td>Language libraries</td>
</tr>
</tbody>
</table>
* RT- POSIX (Real-time standard for making UNIX programs portable)
* OSEK/ VDX (automotive industry, standardization of communication infrastructure)
* ARINC 653
* Micro-ITRON

* Standards for RT OS
* Portable Operating System Interface for UniX (POSIX) is a family of standards.
* RT-POSIX is the real-time extension, it provides:
  * system calls for concurrent programming
  * Task synchronization via mutual exclusion by priority inheritance
  * Task synchronization via condition variables,
  * Data sharing among tasks
  * Prioritized message queues for inter-task communication
Features:

- Fixed priority preemptive scheduling,
- Sporadic server scheduling
- Time management with high resolution,
- Sleep operations,
- Multipurpose timers
- Execution time budgeting
- Virtual memory management
OSEK/VDX is a standard of the automotive industry which addresses safety-critical real-time applications.

Applications are grouped into functions which are mapped to a huge number of resource units.

As main objective the standard defines the communication environment for automotive control unit applications.

It also includes a standardization of the inter-networking interfaces for ensuring safety and reliability of communication networks.

The standard does not achieve portability of the software, as the I/O interaction of the devices is not part of the standard.
ARINC 653 (Avionic Application Standard Software Interface 653) for implementing, integrating and certifying analyzable safety-critical RT applications for integrated modular avionics architectures

Main idea: space and time partitioning, which excluded interference and makes systems predictable.

Each partition represents a different application, with its own memory space and its time slot for accessing a resource, e.g. processor.
* Applications are made of tasks which communicate via message buffers, semaphores and events and executed under fixed priority static scheduling.

* Tasks of different partitions communicate via exchanging messages over ports provided by the API of the underlying OS.

* For task communication it is transparent if the receiver is located on the same processor, a different partition or just an interface device.

* Ports can be used in a sampling mode and a queuing mode. In the former mode not ready messages are overwritten.

* ARINC 653
* Micro/ITRON (Industrial RT OS Nucleus) is a de-facto standard for embedded systems in Japan.

* Standardized kernel functions, called **standard profiles**

* The standard profiles support task priorities, semaphores, message queues and mutual exclusion primitives with priority celling and priority inheritance