Hierarchy of Software Complexity

Application Programs

Sequential Programming

Concurrent Programming

Distributed Programming

Real-Time Programming

Security Programming

Reliable (FT) Programming

Middleware

Embedded Programming

Quality of Service Tradeoffs: accuracy, real-time, security, reliability, device constraints

Autopilot Example

Designate flight plans

Implement the flight plan

Flow the planned trajectory

Device control, e.g., engine
Software Modules Structure

- Displays
- Key Set
- HOTAS
- Store Control

Pilot Interface

- Navigation
- Flight Control
- Mission Computer
- Store management Computer

Computers

- Radar & Infrared
- GPS receiver
- Engines & Actuators

Sensors and actuators

Timing

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Periods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Update</td>
<td>200 ms</td>
</tr>
<tr>
<td>Tracking Target Update</td>
<td>100 ms</td>
</tr>
<tr>
<td>Navigation Update</td>
<td>60 ms</td>
</tr>
<tr>
<td>Data bus polling</td>
<td>40 ms</td>
</tr>
<tr>
<td>Collision Warning Receiver</td>
<td>25 ms</td>
</tr>
</tbody>
</table>
Distributed Scheduling Problem Space Taxonomy

- System
  - Local Only
    - Static
      - All Hard
        - All Periodic
          - Periodic/Sporadic
  - E2E-1 (per request)
  - E2E-N

Identify application and system characteristics

Remote Calls
- Dynamic
  - All Soft
  - Periodicity

System Conditions
- Constraints

Identify application and system characteristics

Priority Driven Example

Precedence Graph

\[
\begin{align*}
J_1, & \quad 3 \\
J_2, & \quad 1 \rightarrow J_3, \quad 2 \rightarrow J_4, \quad 2 \\
J_5, & \quad 2 \rightarrow J_6, \quad 4 \\
J_7, & \quad 4 \rightarrow J_8, \quad 1 \\
\end{align*}
\]

Subscript indicates priority
Number next to job indicates exec time
J_5 released at time 4, all others at time 0
Jobs to be scheduled on two processors: P1 and P2
Priority Driven Example

P1: J1, J4, J7, J6.
P2: J2, J3, J7, J5, J8.

preemptive

non-preemptive

EDF Non-optimality Example

EDF Schedule
Missed deadline

Feasible Schedule
Distributed Scheduling Problem Space Taxonomy

- System
- Local Only
- Static
- All Hard
- All Periodic
- Liu Ch 6
- E2E-1 (per request)
- Dynamic
- Hard/Soft
- All Soft
- Periodic/Sporadic
- All Sporadic
- Liu Ch 7
- E2E-N
- Remote Calls
- System Conditions
- Constraints
- Periodicity
- Liu Ch 9

Identify application and system characteristics

Distributed Scheduling Solution Space Taxonomy

- Scheduling
  - Fixed
  - Offline Only
  - Priority
  - Dynamic
  - Liu Ch 8
  - Resource Allocation

- Adaptive
  - On/Offline
  - Reservation
  - Static
  - Liu Ch

- Adaptive?
  - Online Only
  - Analysis
  - Mechanism
  - Parameters
  - Liu Ch

Distributed scheduling is a collection of algorithms that (currently) use priority or reservation capabilities of underlying subsystems