机器人操作系统及开源软件平台研究

北京航空航天大学
中国电子学会嵌入式专委会

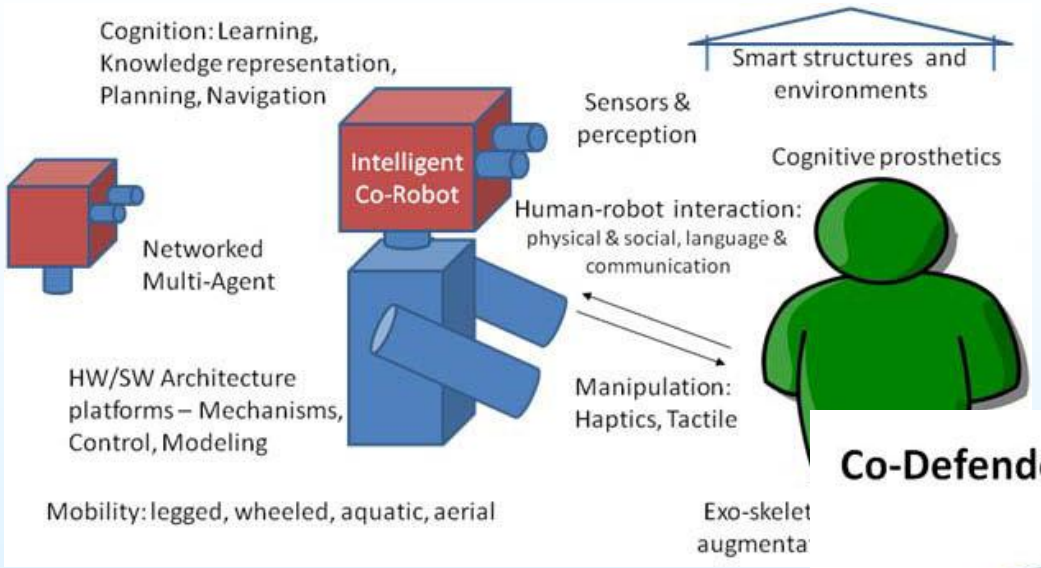
魏洪兴

当前机器人领域面临的挑战

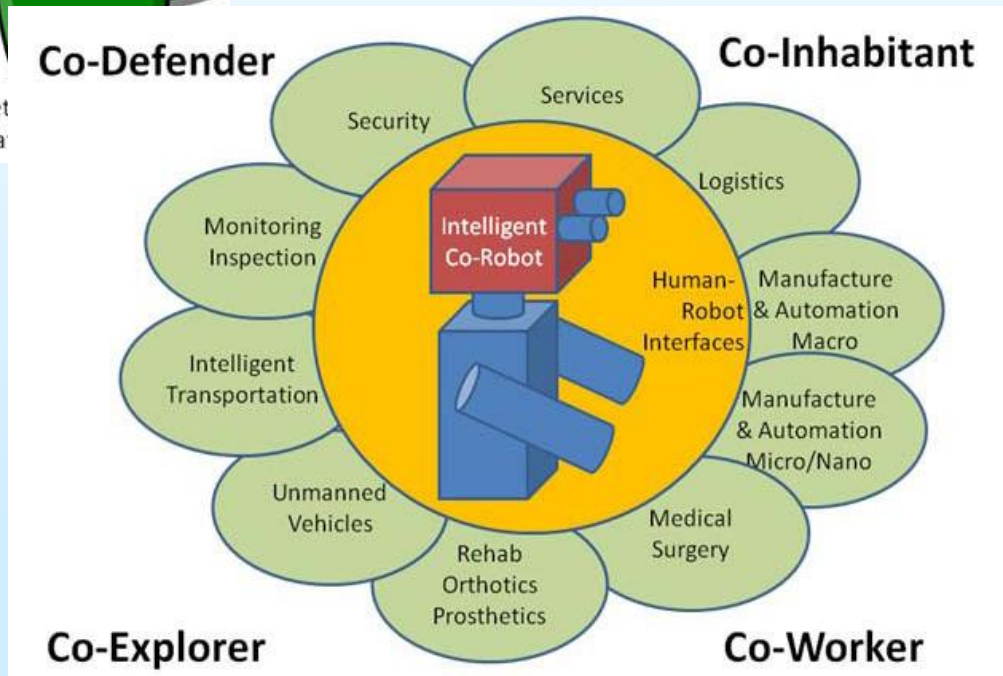
- 多样化的体系结构与硬件平台
- 有限的开发工具与模块
- 技术体系复杂，需要专家知识
- 机器人软件不能有效重用
- 缺乏量大面广的典型应用
- ……

类似早期的PC产业

美国下一代机器人研究计划（NRI）

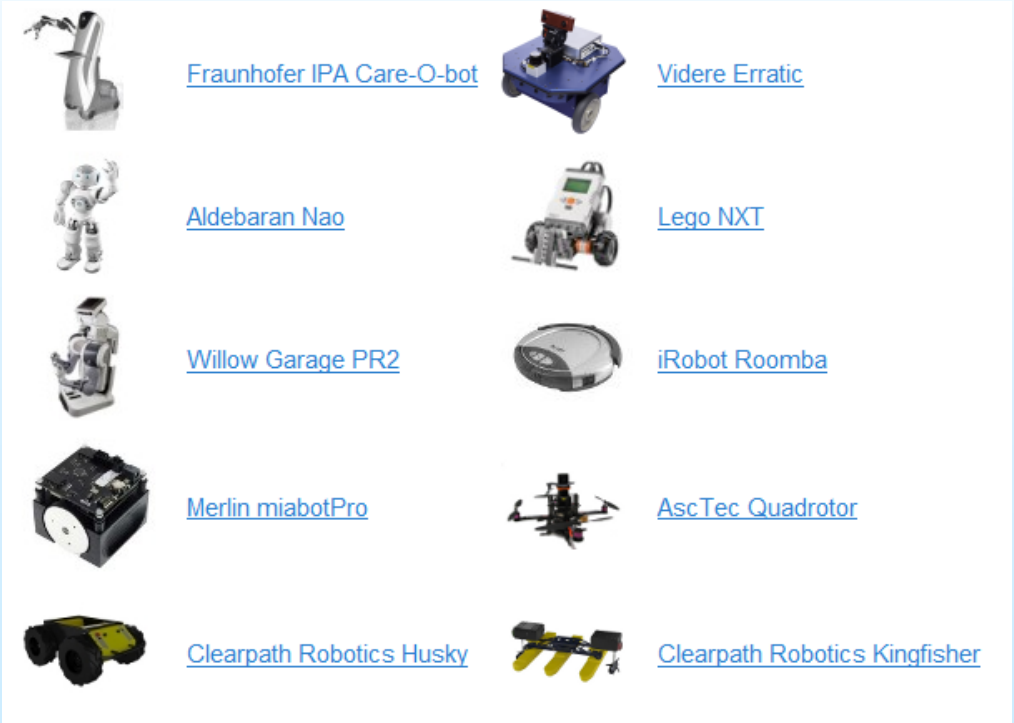


首次提出机器人操作系统及软件平台共享计划



ROS的发展与应用情况

ROS（机器人操作系统，Robot Operating System），是专为机器人软件开发所设计出来的一套操作系统架构，它能够支持多种机器人构型和传感器。它是一个开源的操作系统，包括硬件抽象描述、底层驱动程序管理、共用功能的执行、程序间消息传递、程序发行包管理，它也提供一些工具和库用于获取、建立、编写和执行多机融合的程序。ROS的能够提高机器人研发的代码复用率，集成与其他机器人软件框架。

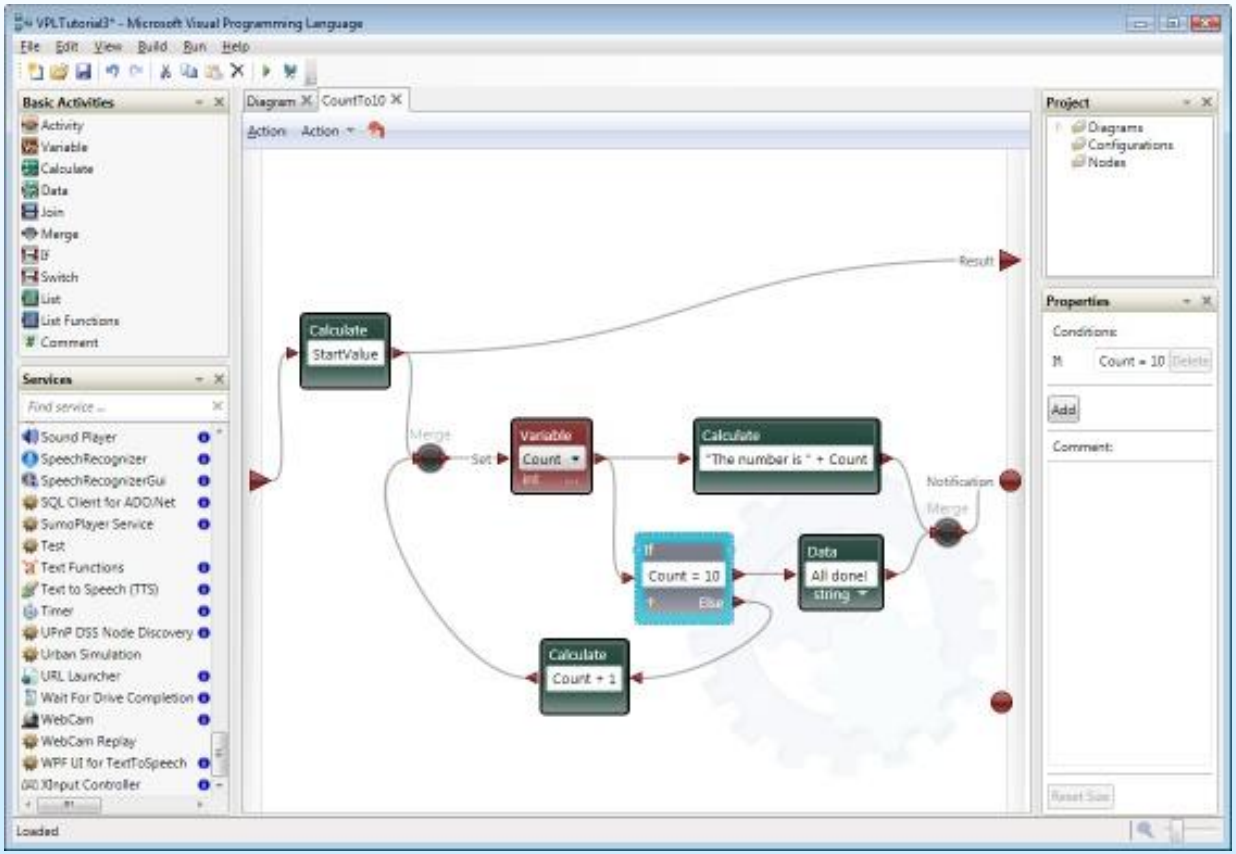


2007年，ROS最初由斯坦福人工智能实验室的Switchyard为了支持斯坦福人工智能机器人（STAIR）项目而开发的。

2008年，ROS的主要由Willow Garage公司继续开发，Willow Garage是一家机器人技术研究机构和机器人技术孵化机基地。

微软Microsoft Robotics Developer Studio（MRDS）

Microsoft Robotics Developer Studio（MRDS）是一个给“学术研究者、爱好者和商业开发者创建涵盖广泛硬件平台的机器人应用程序”的Windows环境。MRDS主要用来在3D物理虚拟环境中模拟机器人应用程序，并使用Windows或Web的接口和机器人进行交互。



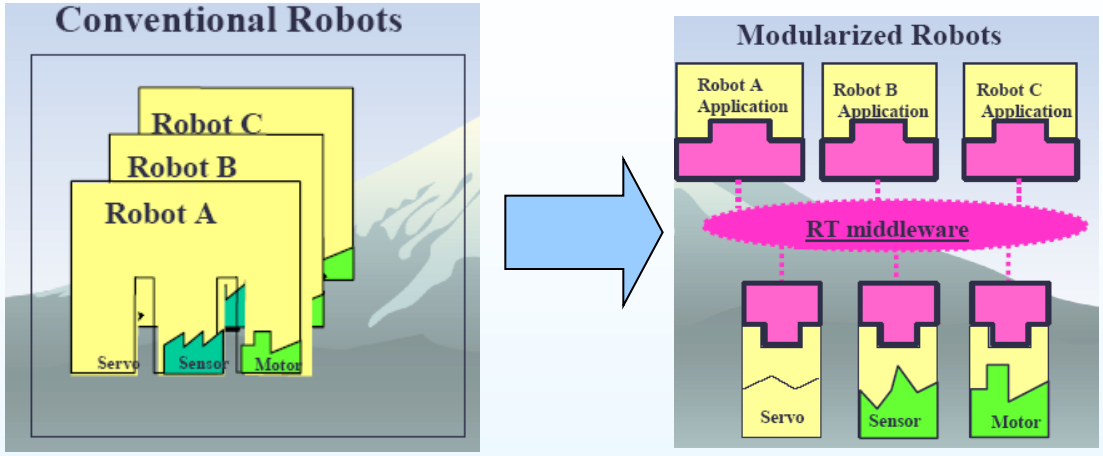
机器人操作系统的要素分析

- 实时性
- 丰富的软件资源：运动学、动力学、路线规划、导航、SLAM、人机交互
- 集成开发环境：降低使用难度

典型操作系统的实时分析

- 通用操作系统：Linux, Windows
  - 优点：编程资源丰富
  - 缺点：实时性差
- 实时操作系统：uC/OS, vxWorks
  - 优点：实时性好
  - 缺点：编程资源贫乏
- 混合操作系统：同时运行通用操作系统和实时操作系统

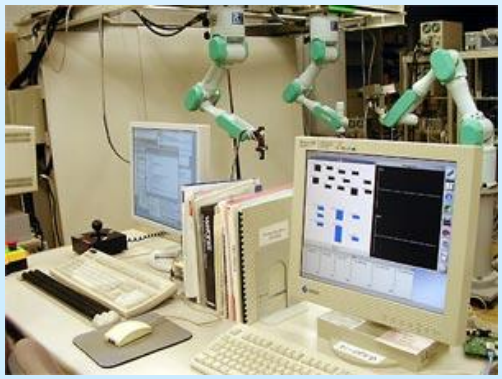
日本“OpenRTM-aist”机器人中间件标准化体系设计



Concept of RT middleware Concept

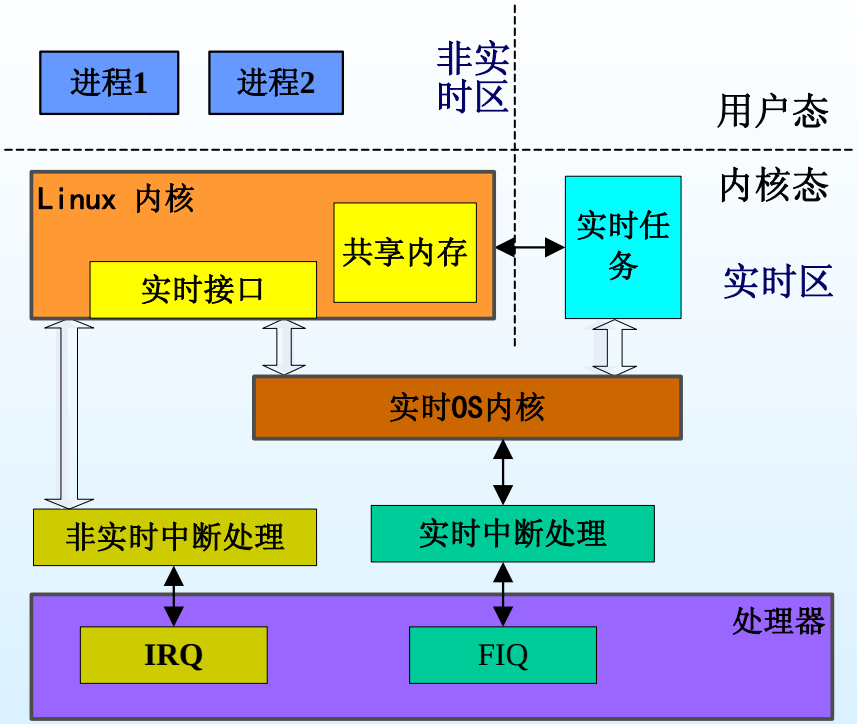
- **An Open interface**
- **Modularized functional components**

OpenRTM-aist采用了CORBA（Common Object Request Broker Architecture,公共对象请求代理体系结构）的体系结构，AIST（National Institute of Advanced Industrial Science and Technology，日本国家高级产业科学技术研究院）在OMG组织成立了机器人工作组（Robotics Domain Task Force），目的是通过采用OMG标准加快机器人系统从模块组件级的集成。

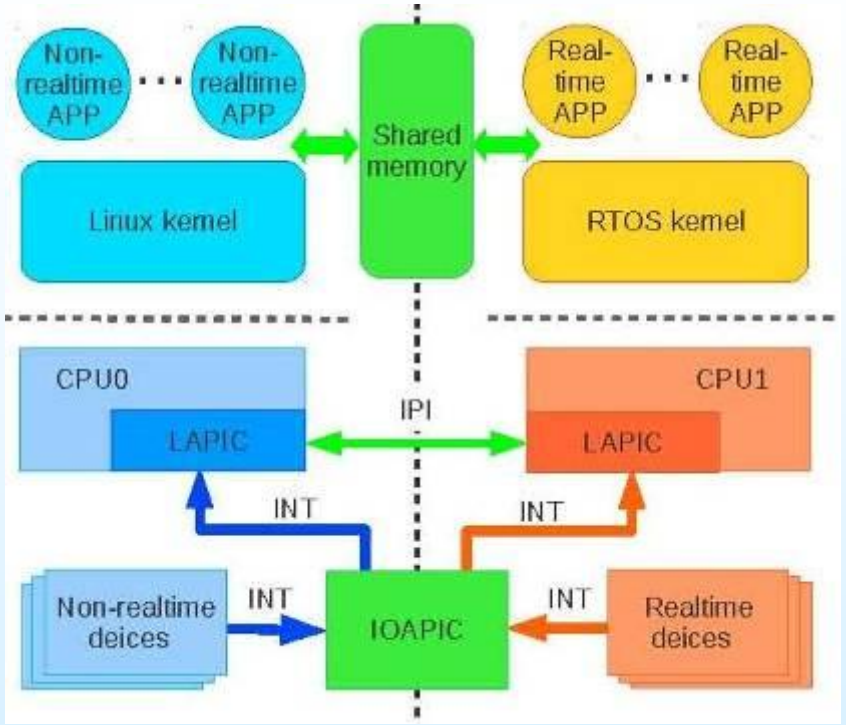


使用模块化功能构件和相同控制软件的工业机器人(左)和仿人机器臂(右)

机器人实时操作系统RobOS（双核操作系统）



RobOS的体系结构

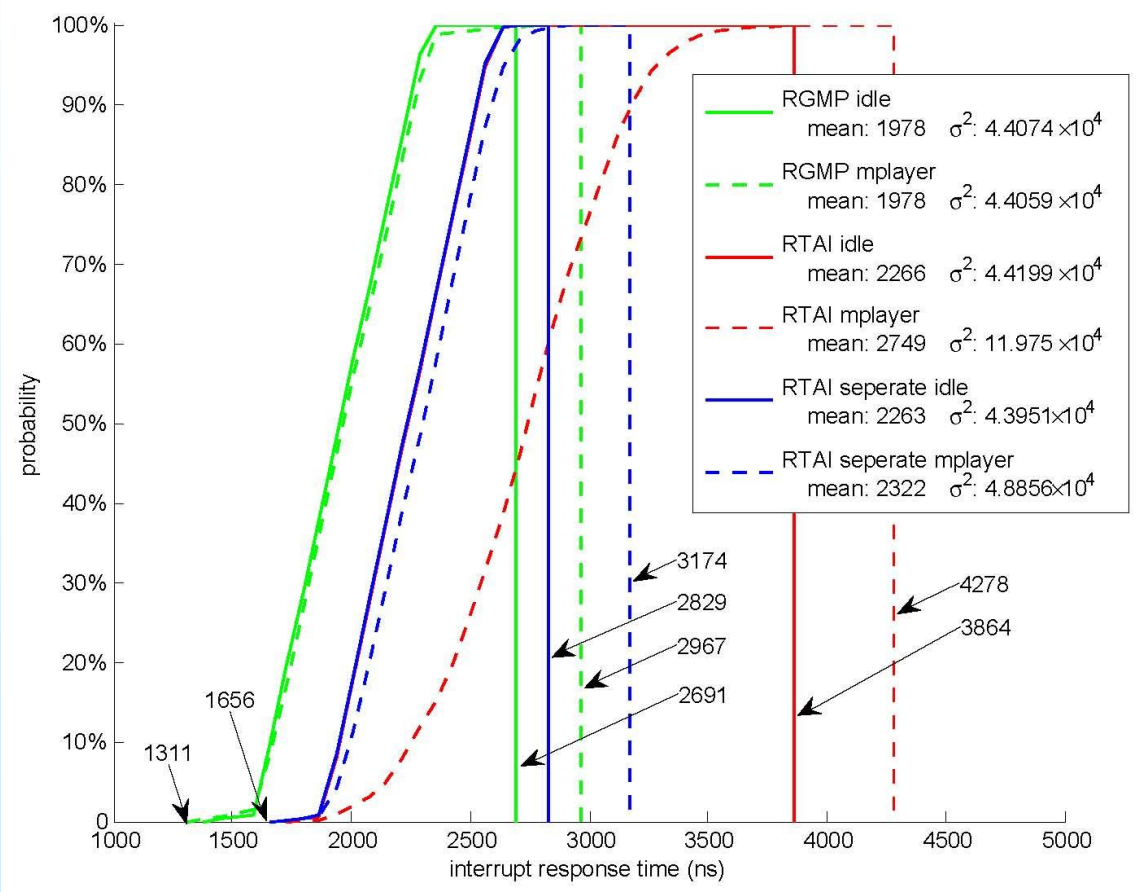


RobOS在多核处理器上的应用体系

On Improving Real-Time Interrupt Latencies of Hybrid Operating Systems with Two-Level Hardware Interrupts Liu, M; Liu, D; Wang, Y; et al. *IEEE TRANSACTIONS ON COMPUTERS* 60 (7): 978-991 JUL 2011

RobOS实时性能分析

- 中断响应时间对比试验



实时化操作系统内核

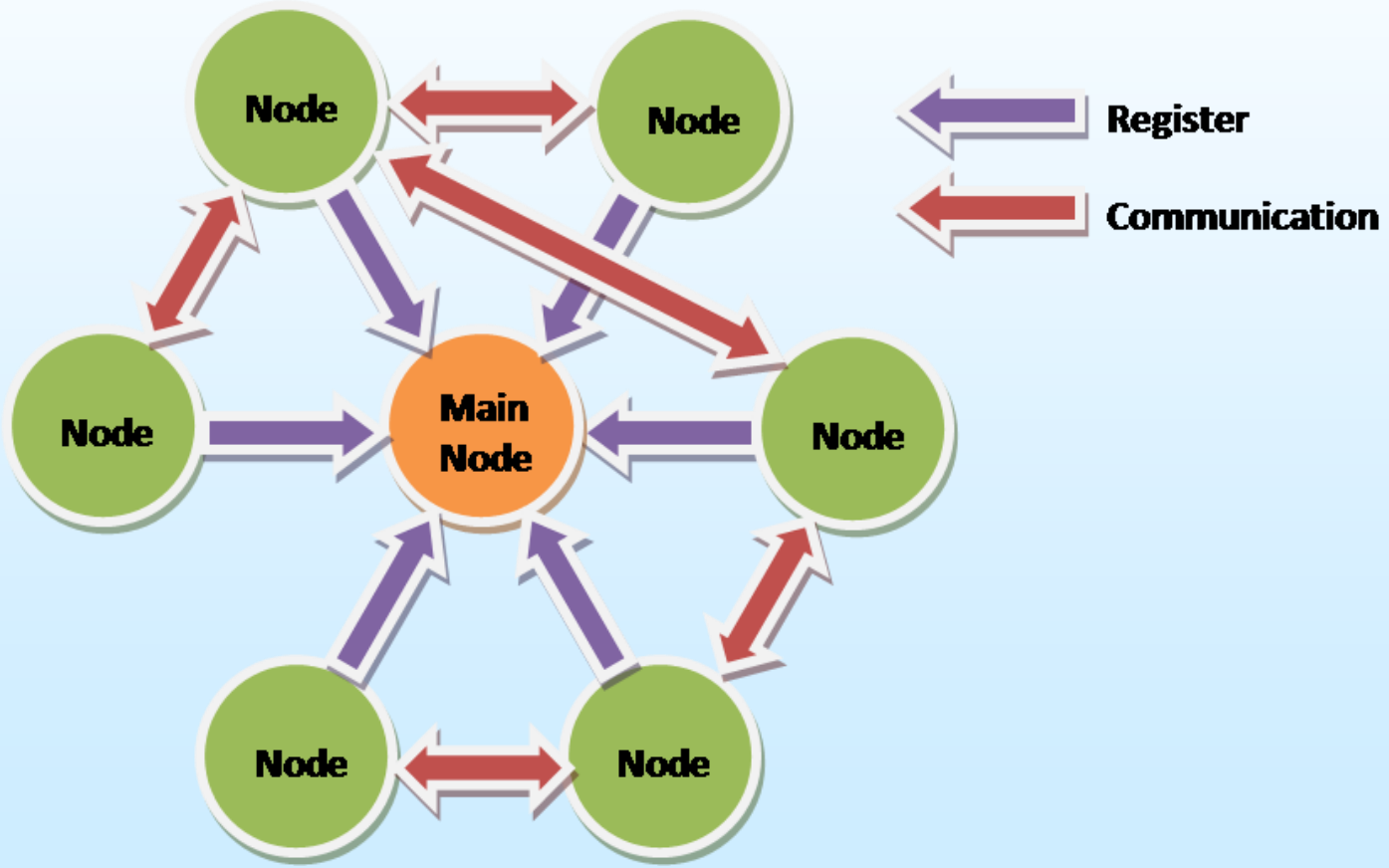
- 实时操作系统移植：
  - uC/OS
  - NuttX

实时化操作系统内核

- RGMP项目网站：http://rgmp.sf.net
  - 开源发行版
  - BSD许可证
  - 使用、开发文档

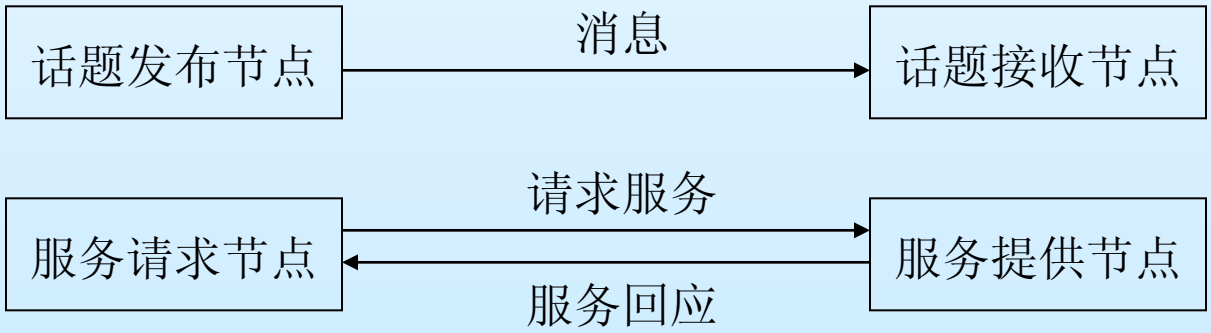
基于机器人操作系统的分布式软件体系结构

- 分布式体系架构



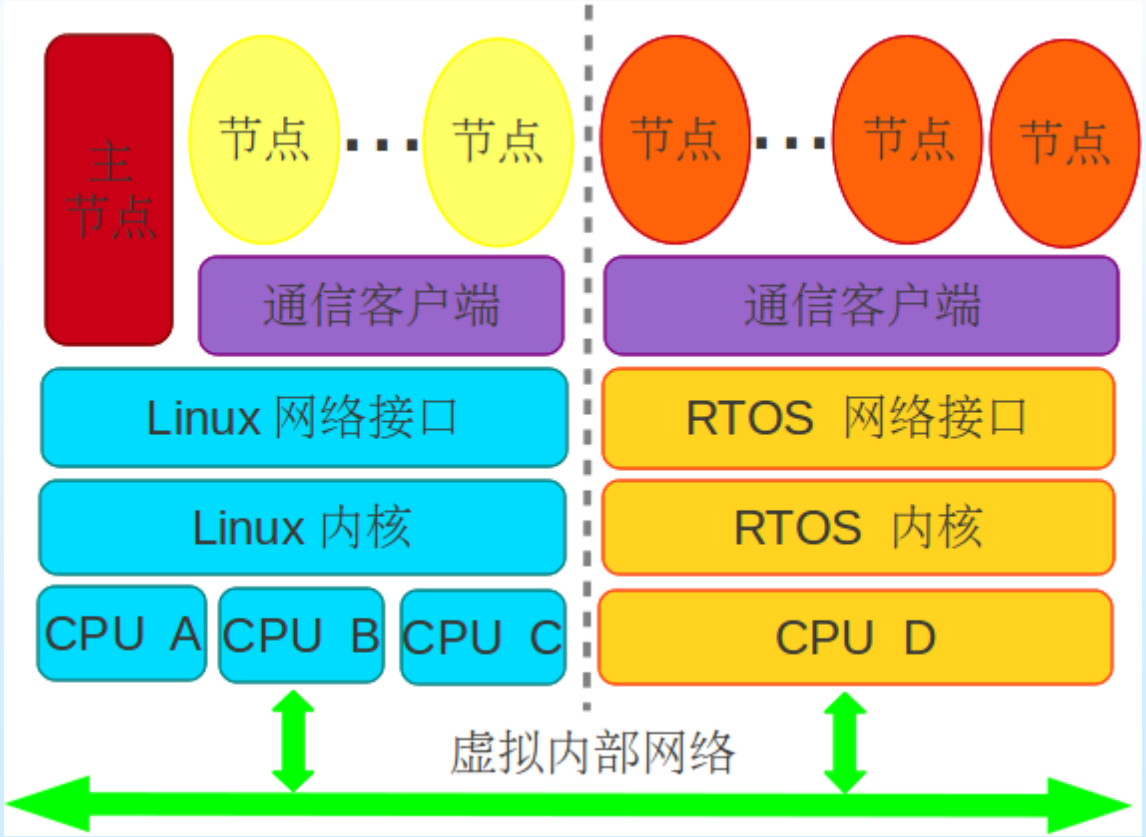
机器人分布式软件实现原理

- 节点：分布式的进程
- 服务：同步通信
- 话题：异步通信
- 消息：通信数据结构



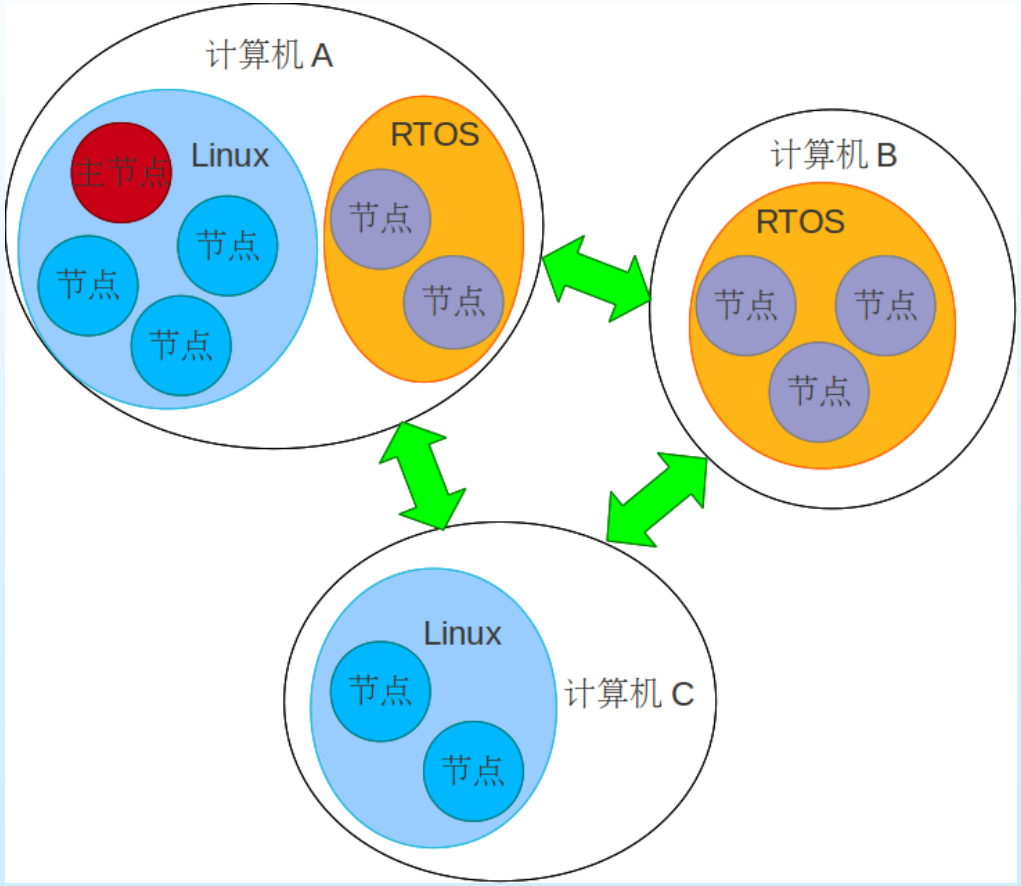
基于操作系统的应用

- 与RGMP结合（实时化）

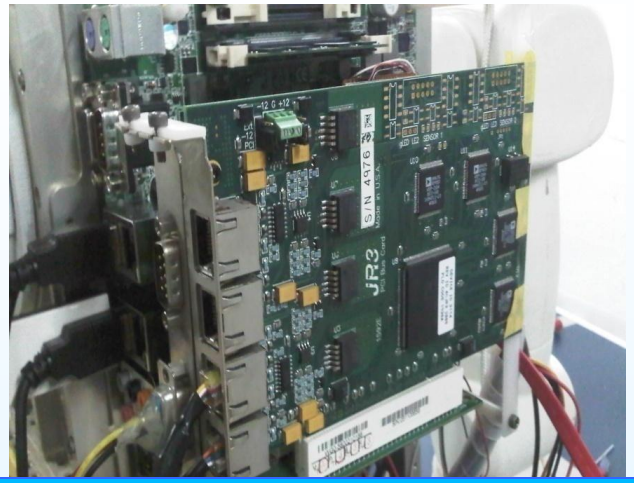


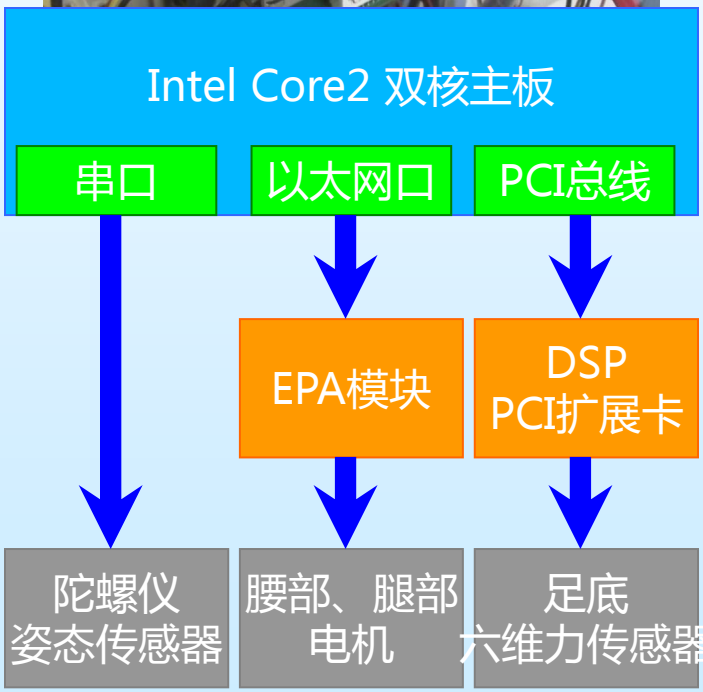
机器人操作系统

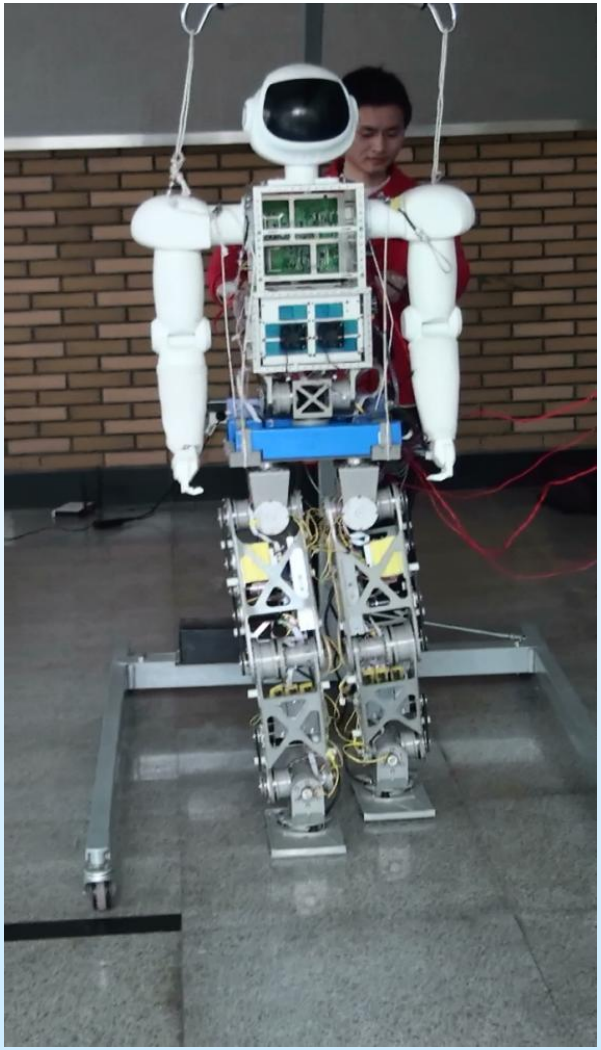
- 与RGMP结合（实时化）



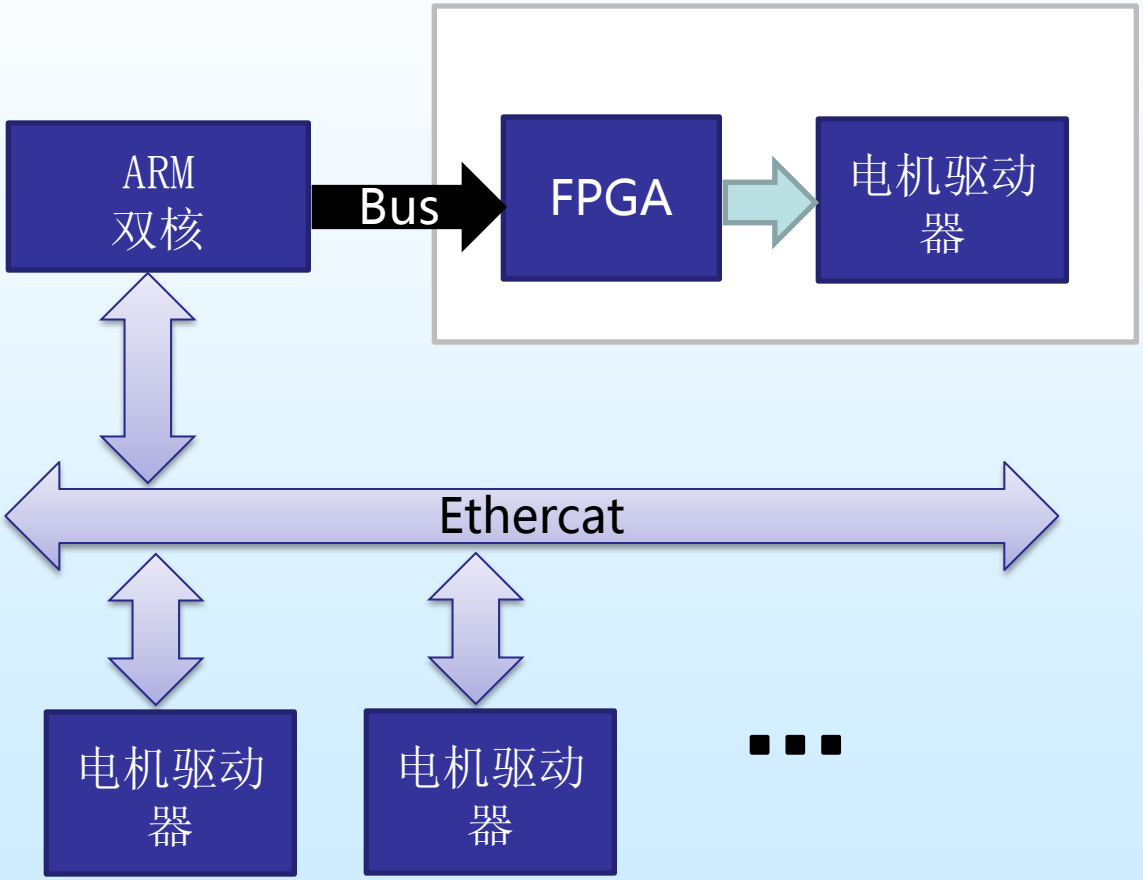
机器人操作系统在仿人机器人上的应用





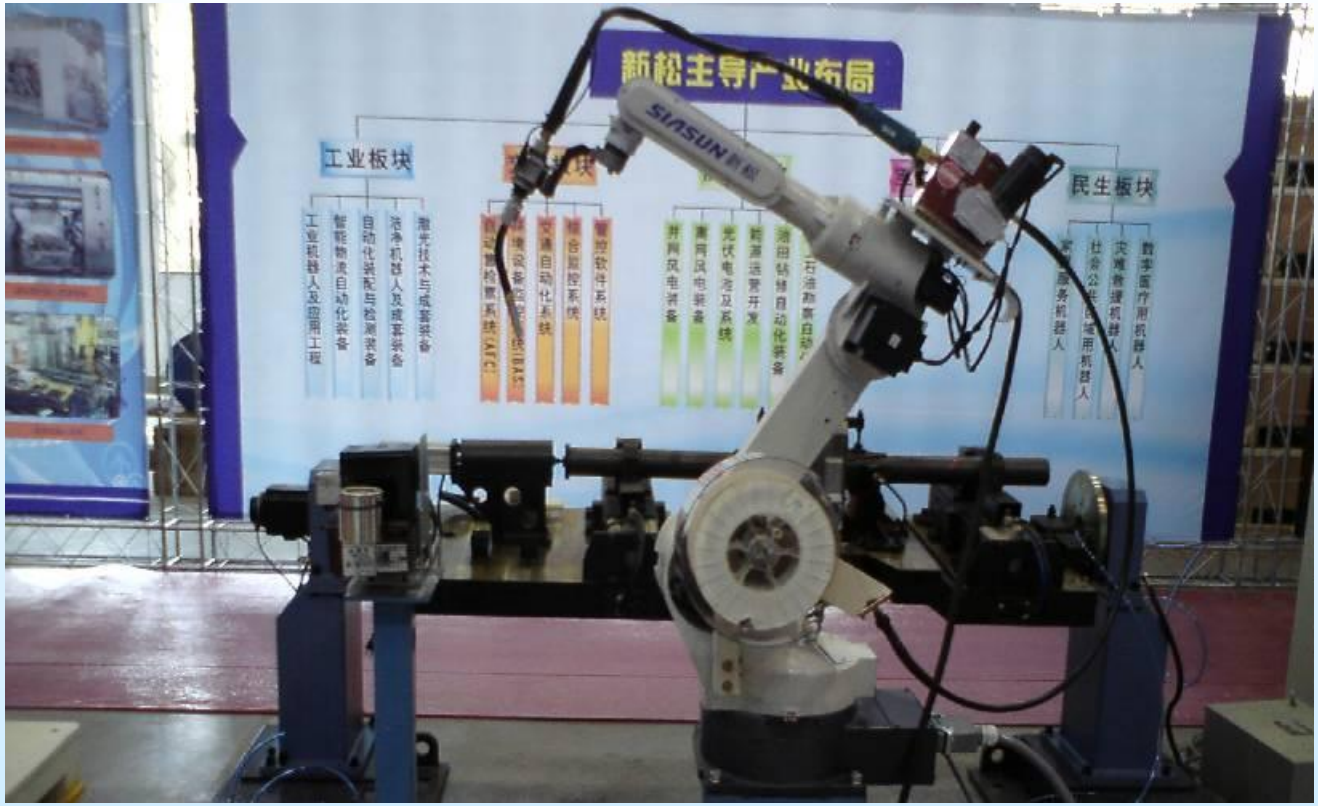


机器人操作系统在工业机器人控制器的应用



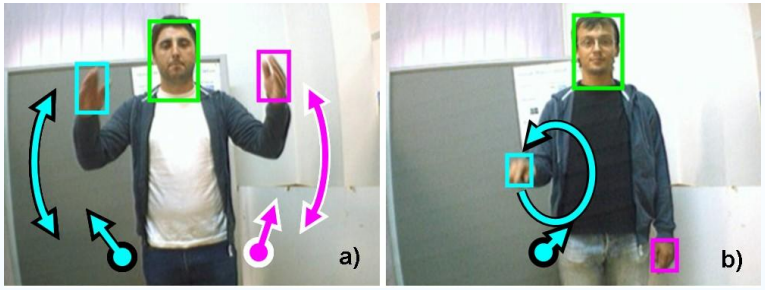
应用于弧焊机器人

新松6kg弧焊机器人



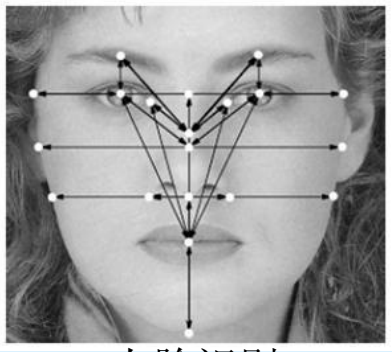
构建机器人软件仓库

运动重建结构
（Structure from motion）



手势识别
（Gesture Recognition）

运动跟踪
（Motion Tracking）



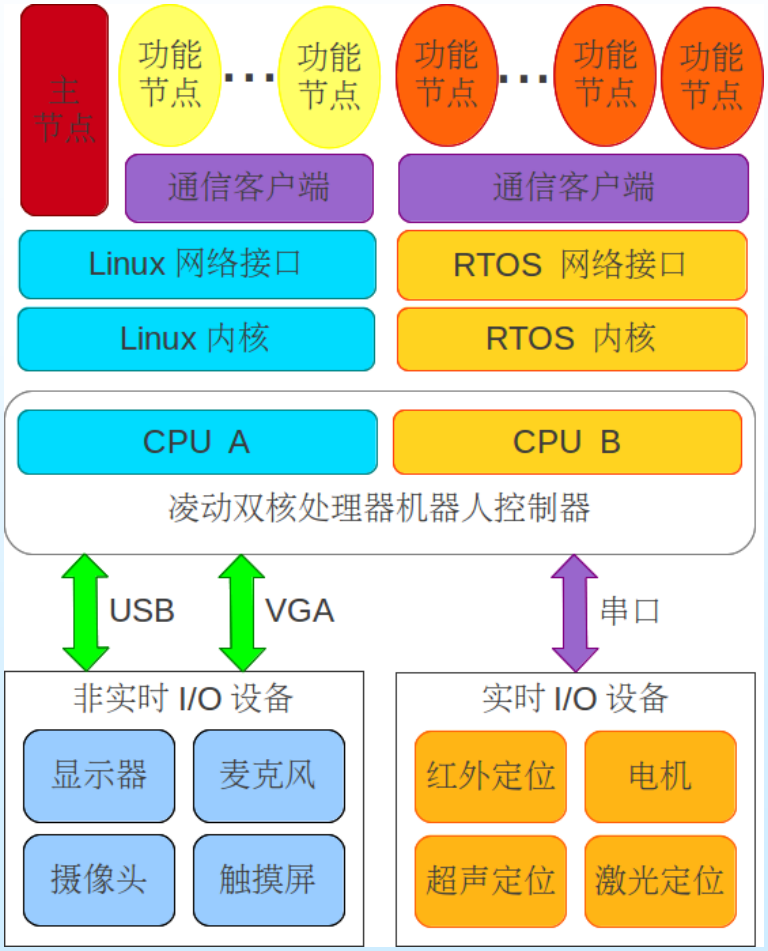
人脸识别
（Face Recognition）

其他算法：
Object Identification（目标识别）、Segmentation and Recognition（分割与识别）
Mobile Robotics control（移动机器人控制）、Motion Understanding（运动理解）
Stereopsis Stereo vision: depth perception from 2 cameras（双目立体视觉）
Planning（路径规划）

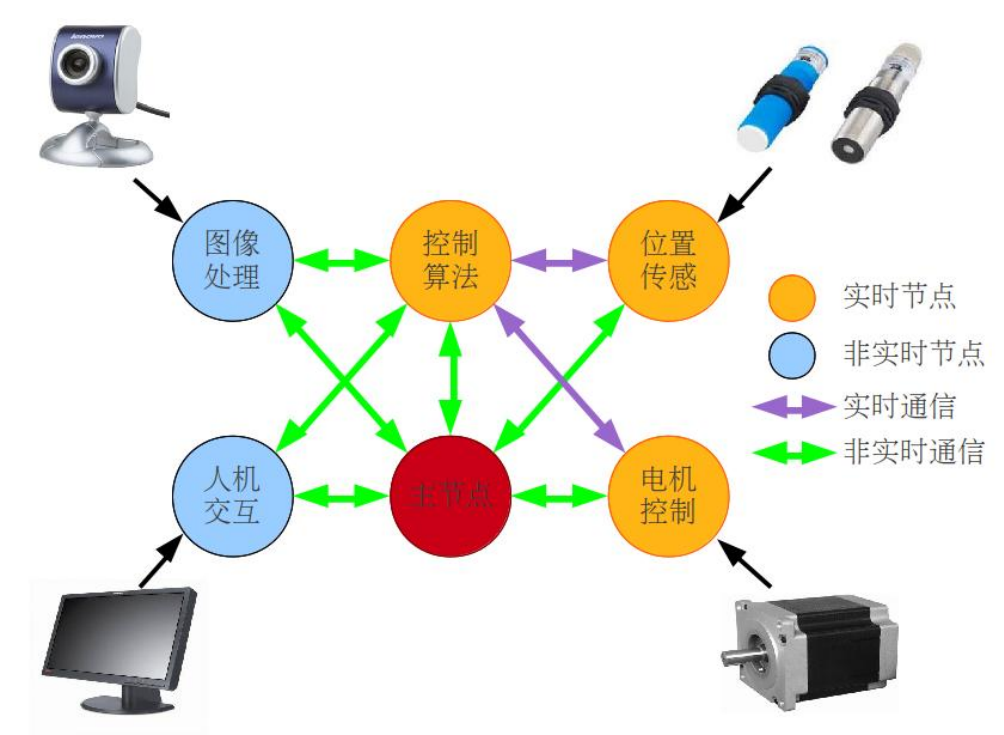
建立集成开发环境

- 使用Eclipse插件开发技术
- 提供可视化编辑功能
- 完善的工程管理功能
- 图形化导入功能节点和自定义节点
- 简单快速地实现机器人特定的应用。

医疗陪护机器人的应用案例



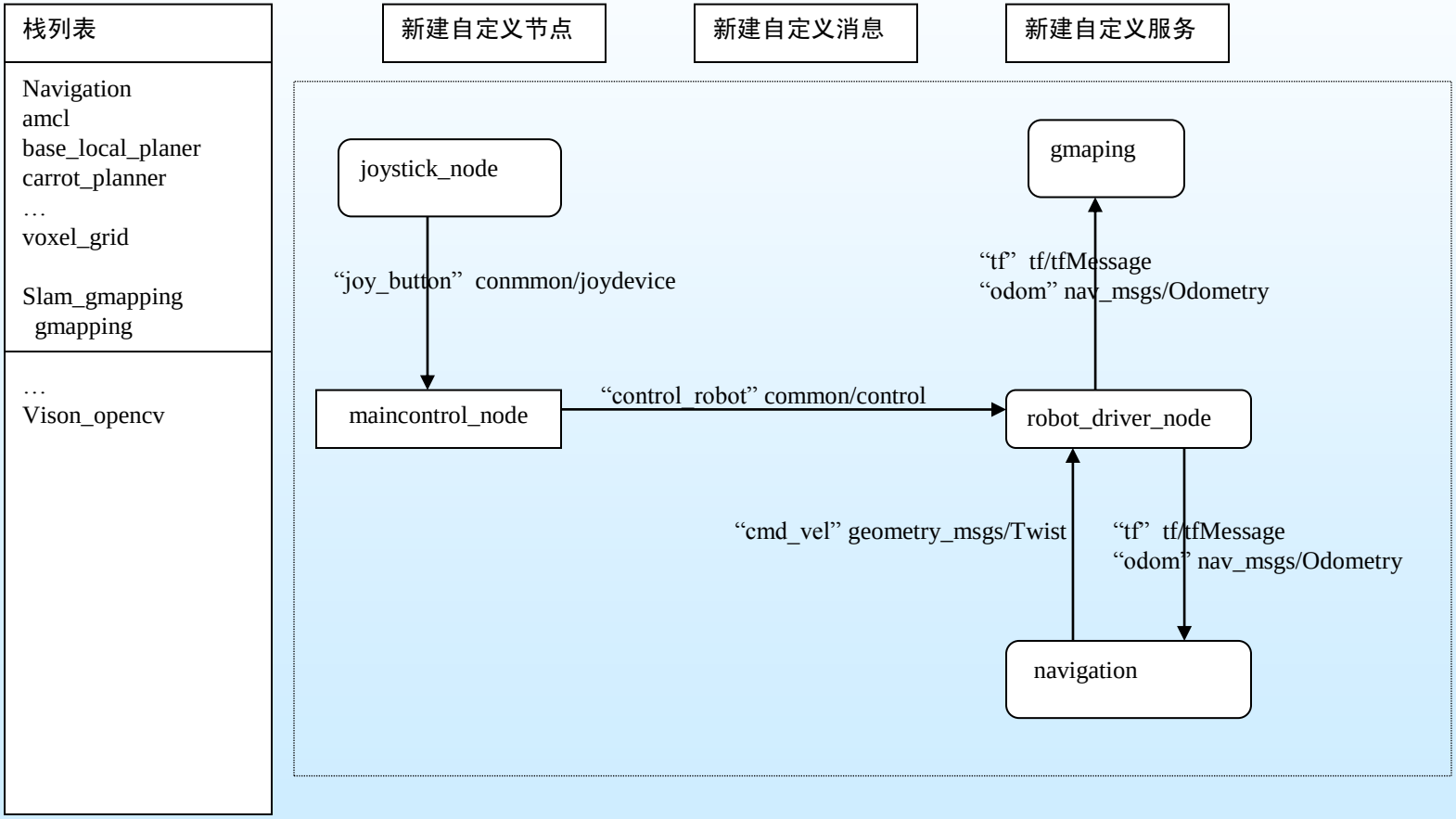
医疗陪护机器人的应用案例



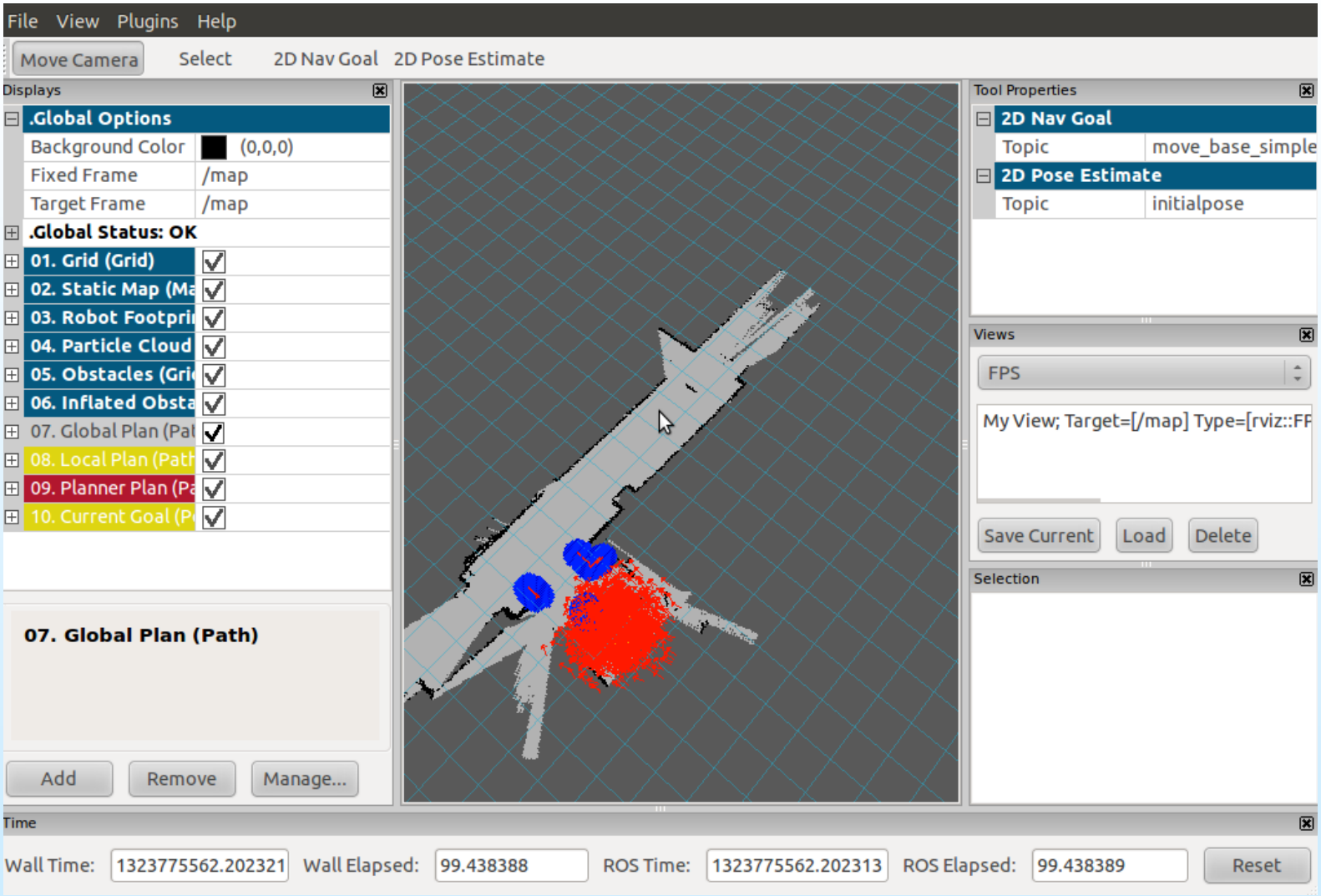
SLAM导航开发流程

- 新建工程SLAM-NAVIGATION。
- 选择系统提供功能节点，拖入工作区。
  - 地图功能节点gmapping。
  - 导航功能节点navigation。
  - 手柄控制节点joystick_node。
  - 机器人电机驱动节点robot_driver_node，
- 新建自定义节点maincontrol_node，该节点提供图形化控制界面。
- 建立节点间消息路径。
  - 连接gmapping节点和robot_driver_node节点，实现robot_driver_node向gmapping发送“tf”坐标系转换和“odom”机器人里程信息。
  - 连接navigation节点和robot_driver_node节点，实现robot_driver_node向navigation发送“tf”坐标系转换和“odom”机器人里程信息。
  - 连接navigation节点和maincontrol_nod节点，实现navigation向robot_driver_node发送机器人速度信息“cmd_vel”。
  - 连接joystick_node节点和maincontrol_node节点，实现joystick_node向maincontrol_node发送手柄事件信息。
  - 连接robot_driver_node节点和maincontrol_node节点，实现robot_driver_node接受maincontrol_node发送的控制机器人命令信息。
- 编辑节点maincontrol_node的代码
- 编译运行

SLAM工程结构


栈列表
Navigation
amcl
base_local_planer
carrot_planner
…
voxel_grid
Slam_gmapping
gmapping
…
Vison_opencv
新建自定义节点
新建自定义消息
新建自定义服务
joystick_node
gmaping
“joy_button” conmmon/joydevice
“tf” tf/tfMessage
“odom” nav_msgs/Odometry
maincontrol_node
“control_robot” common/control
robot_driver_node
“cmd_vel” geometry_msgs/Twist
“tf” tf/tfMessage
“odom” nav_msgs/Odometry
navigation

SLAM工程建立地图


File View Plugins Help
Move Camera
Select
2D Nav Goal
2D Pose Estimate
Displays
.Global Options
Background Color
(0,0,0)
Fixed Frame
/map
Target Frame
/map
.Global Status: OK
01. Grid (Grid)
02. Static Map (Ma
03. Robot Footpri
04. Particle Cloud
05. Obstacles (Gri
06. Inflated Obsta
07. Global Plan (Pat
08. Local Plan (Path
09. Planner Plan (Pa
10. Current Goal (P
07. Global Plan (Path)
Add
Remove
Manage...
Tool Properties
2D Nav Goal
Topic
move_base_simple
2D Pose Estimate
Topic
initialpose
Views
FPS
My View; Target=[/map] Type=[rviz::FF
Save Current
Load
Delete
Selection
Time
Wall Time:
1323775562.202321
Wall Elapsed:
99.438388
ROS Time:
1323775562.202313
ROS Elapsed:
99.438389
Reset

Q&A

weihongxing@buaa.edu.cn

2011.12.17