**题目：扩展的内存管理机制设计实验**

**目的：**在了解实时嵌入式操作系统内存管理机制的特点以及实时处理对内存管理需求的基础上，练习并掌握有效处理内存碎片的内存管理机制，同时理解防止内存泄漏问题的良好设计方法。

**环境：**VxSim仿真环境或TQ2440(ARM920T)

**实验原理及要点（知识点）：**

实时多任务操作系统（如VxWorks）为保证其时实性，在内存管理方面往往采用最简单有效的方式进行（如malloc/free），但同时会辅以扩展功能（如内存分区-Memory Partition、Cache安全等）以满足用户对内存操作的基本需求。

但实践者应该认识到，在使用操作系统所提供的内存管理机制时，需注意防止其造成的潜在危险，需要形成良好的使用习惯，同时根据应用的特点适当扩展。其关键点在于：

1. 实时系统的内存管理的主要目标为：快速、高效、高可靠；

2. 内存碎片（外部碎片）是危害系统稳定的重要原因之一；

3. 预先规划是达到以上目标的一种良好手段。由于实时系统无法使用复杂的内存管理机制（如垃圾回收等），用户可以根据实时应用使用内存的特点，扩展自己私有的内存管理机制；

4. 内存泄漏的防治主要依赖于内存使用上严谨的编程习惯。

**内容：**

1. 使用预先规划的思想，构建自己的私有内存管理机制，在系统内存池中申请内存，并将其纳入私有内存管理机制中，形成静态预分配内存池；

2. 静态预分配内存池支持一种以上固定长度内存池，如16字节内存池和256字节内存池。固定长度内存池的单块长度应考虑体系结构开销，并尽量减少内部碎片；固定长度内存池数量应可配置；

3. 静态预分配内存池与系统内存池的统一管理机制。向用户分配内存时应保证长度最佳匹配原则。当申请内存的长度超过静态预分配长度或资源不足时，自动向系统内存池申请；

4. 管理机制至少应包括：

a) 初始化函数；

b) 内存申请/释放函数。并特别要保证释放安全；

c) 管理监视机制。

5. 利用可能的互斥机制或代码可重入设计，保证以上管理机制的操作安全性；

6. 创建多Task环境测试及演示以上内容。

**提交：**

1. Workspace：含project及运行说明。代码注释不少于30%。

2. 实验报告。框图说明系统结构，给出系统测试运行的典型WindView图，文字说明系统各部分定义细节。

3.以上内容提供一个压缩文件，文件名：学号-姓名-实验名称。