Chapter 10

Robust Concurrency and Synchronization for Large-Scale Medical Robot Operations

Qicheng Li
http://orcid.org/0009-0008-3709-9290
The Hong Kong Polytechnic University, Hong Kong

Fenggang Yu
The Hong Kong Polytechnic University, Hong Kong

Chengyang Li
The Hong Kong Polytechnic University, Hong Kong

Shengjie Zeng
The Hong Kong Polytechnic University, Hong Kong

Chengjun Yu
The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

Rapid AI advances have enabled autonomous healthcare robots to perform delivery and disinfection, yet coordinating large-scale operations remains challenging. To prevent management chaos, our team developed a medical-robot coordination system focused on concurrency control, synchronization, and coordination. The chapter implements three core and ten advanced features, including task allocation and

DOI: 10.4018/979-8-2600-1101-0.ch010

real-time monitoring, and benefits from modular development and collaborative debugging. Each team member contributed to key features, testing, and comprehensive documentation, yielding a robust, well-documented system.

1. INTRODUCTION

1.1 Problem Statement and Motivation

1.1.1 Problem

As embodied intelligence and robotic science rapidly develop, many international modern hospitals have inserted autonomous medical care robots into the medical care system (Deepika et al., 2026). However, it is an essential problem that a highly efficient OS core is needed to coordinate numerous robots in hospitals. Therefore, a construction of medical robots OS core with appropriate concurrency, synchronization, and safe coordination must be put on the agenda.

To effectively address this problem, this paper constructs an OS core for medical robots that can coordinate and manage them in a unified manner. The OS core is aiming to clearly and systematically schedule all the medical care robots within the system to handle tasks of various priorities and different sequences in an orderly fashion.

1.1.2 Motivation

We have learnt a lot of knowledge about CPU scheduling, concurrency, deadlock fixing and detecting, and process synchronization. To practice more theory knowledge, we want to integrate and apply all the knowledge we have learned.

At present, the medical robots in hospitals mainly handle routine tasks and assist doctors in their work. The tasks they are responsible for have certain levels of priority. At the same time, collaboration in order within tasks of the same priority should be guaranteed, and starvation should be avoided. At most, only one medical robot works in the same area. Otherwise, it is prone to conflicts on the one hand and will interfere with each other on the other hand. What's more, if the medical robot is offline for too long or gets stuck in a task, in order to prevent area occupation and waste of ineffective resources, it is necessary to monitor its operation status in real time and set a shutdown method (Mohammed & Zangana, 2026).

To handle these requests, we built up an OS core through PR + FIFO task queue, healthy monitor based on heartbeat parameter, and zone occupy.

Related Content