



# Implementing Multilevel Queue Scheduler

In xv6

# Introduction

- This project focuses on implementing multilevel queue scheduler in monolithic kernel xv6.
- Xv6 is a teaching operating system which is used to understand and implement operating system concepts.
- Xv6 is written in ANSI C programming language and is easy to modify.
- You can run xv6 using emulators like qemu.
- We use qemu as it is lightweight and it takes seconds to compile the OS XV6.

# Introduction

- Multilevel queue scheduling is a scheduling policy used to schedule processes by OS.
- Multilevel queue scheduling consists of different levels for processes.
- Each level may have different scheduling policy and may get different proportion of CPU time.
- Eg. Background processes can have First come first serve scheduling while foreground processes can have round robin scheduling.
- This project modifies the xv6 kernel which uses round robin scheduling to use multilevel queue scheduling.

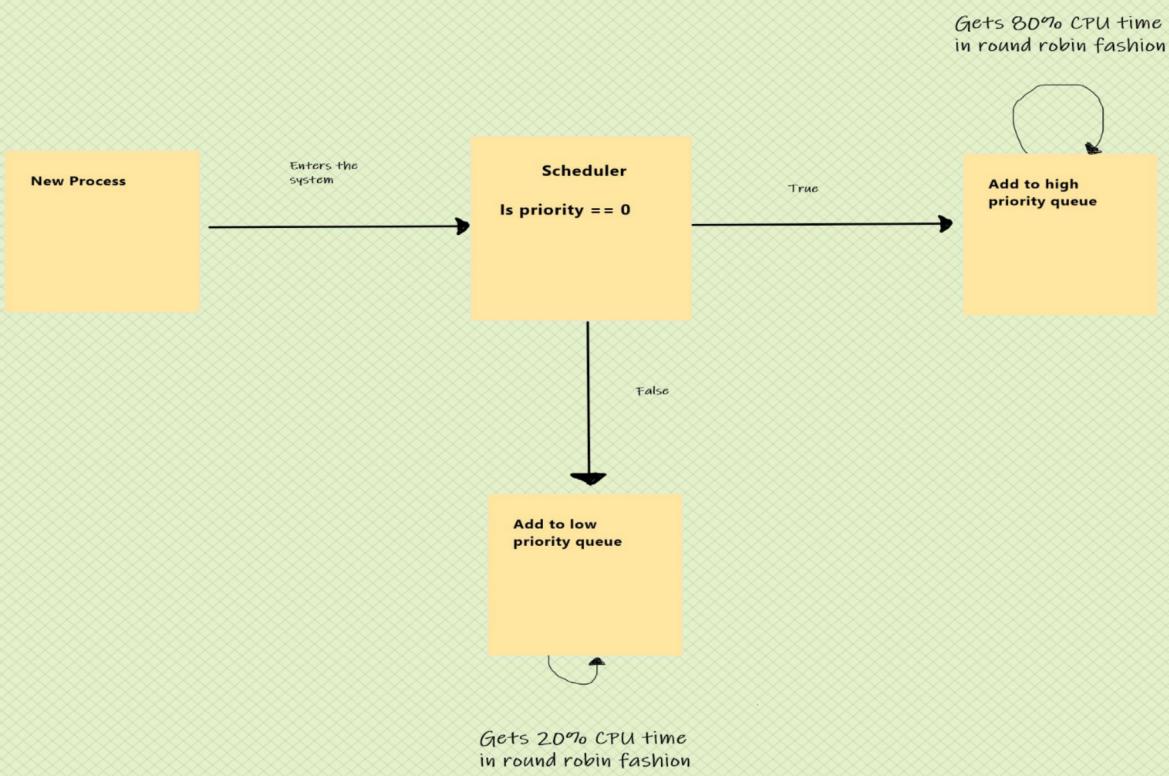
# Motivation

- Designing a scheduling policy has its roots in decision making.
- A bad decision may cause severe damage.
- Making right decision is challenging.
- So I wanted to take up this challenge which is the main motive behind the project.
- Maintaining concurrency of a computer systems involves the important task of scheduling.
- Without scheduling, the interactivity of today's operating system would have been a dream.
- Hence it is extremely important to choose/design a good scheduling scheme.

# Project Goals

- Learn how different OS concepts are implemented using xv6
- Learn about the intricacies of operating system
- Implement a scheduling scheme taking into account scheduling criteria
- Learn how to implement a system call and user programs in xv6
- Implement a scheduling scheme that avoid the problem of starvation

# Design Architecture of Multilevel queue scheduler



# Design Architecture

- To implement this scheme I used queue implemented with linked list.
- I modified Process control block of xv6 processes to include a field for priority.
- I needed a system call to change the priority of a process and a program to create process from shell to illustrate this scheme.

## Outcome

- Outcome of this project is a scheme which is starvation free as it uses round robin scheme as base scheme for both queues.
- This scheme promises fairly high CPU utilization with satisfactory throughput.
- Waiting time for processes in high priority queue is comparatively less than processes in low priority queue.
- Turnaround time is low for high priority processes than for low priority process
- This scheme has high response time.
- Thus this scheme is a good scheme in accordance with scheduling criteria.