

EE3450 2023 Fall  
Computer Architecture  
Program Assignment 1

# Assignment Description

- For this assignment, you have to write an assembly code in RISC-V to implement **QuickSort**.
- After the code is completed, you will learn how to use a toolchain to compile codes, and how to use an ISA simulator to debug.

# Environment Setup

- In the beginning, we have to setup the environment with RISC-V toolchains and ISA simulator (spike).
- TA has provided an VM image with tools listed above.
- Please refer to “***environment\_setup.pptx***” for detailed descriptions.

# RISC-V tools and ISA simulator

- We use RISC-V tools (including compiler, linker, etc.) to build the project and generate executable binaries.
- The ISA simulator runs the compiled binaries. We can use it to check the execution result and debug.
- Please refer to “*run\_tutorial.pptx*” for detailed descriptions.

# Program Assignment 1: QuickSort

- QuickSort is a widely-used sorting algorithm with a time complexity of  $O(n \log_2 n)$ .
- Please understand QuickSort first.
  - [Wikipedia – QuickSort](#)
- The example code in C is provided in the template.

# Program Assignment 1: QuickSort

- In this assignment, you have to finish your code in “\*.S” file for the following 3 parts:
  - Part1: Swap
  - Part2: Partition
  - Part3: QuickSort
- Template code with print helper function are provided by TA. Please do not modify the print function part in the code.

# Part 1 : Swap

- Finish the code “ee3450\_pa1/part1/main.S”.
- For a given array, the program exchanges the value of the first two elements in the array.
- Run and check the correctness of your program.

# Part 2 : Partition

- Finish the code “ee3450\_pa1/part2/main.S”.
- In this part, you have to implement the partition function in the QuickSort algorithm. We select the last element of an array as pivot.
- Run and check the correctness of your program.
- Hint: You can reuse your code in part 1.



# Part 3 : QuickSort

- Finish the code “ee3450\_pa1/part3/main.S”.
- In this part, you have to implement the QuickSort algorithm.
- Run and check the correctness of your program.
- Hint: You can reuse your code in part 1 & 2.

# Delivery

- Rename your main.S in three parts as

- PA1\_<student\_ID>\_part1.S
- PA1\_<student\_ID>\_part2.S
- PA1\_<student\_ID>\_part3.S

For example: PA1\_109061585\_part1.S

- Submit your code directly through eeclass.
- Reminder: Make sure your program works well for other test patterns.

# Note

- Supported input value range for print function: [0, 99]
- Hint: Properly comment your code could make debugging easier.
- Hint: You can refer to the provided C code first.
- Plagiarism is strictly prohibited, including looking at other's work or copying code from the net. TAs will check the plagiarism by programs.