

Anthony Rosenhamer

adros@iastate.edu

CPR E 458 Section 1

Chris Lopez

cllopez@iastate.edu

Project Proposal: Uniprocessor Scheduling

Project Type

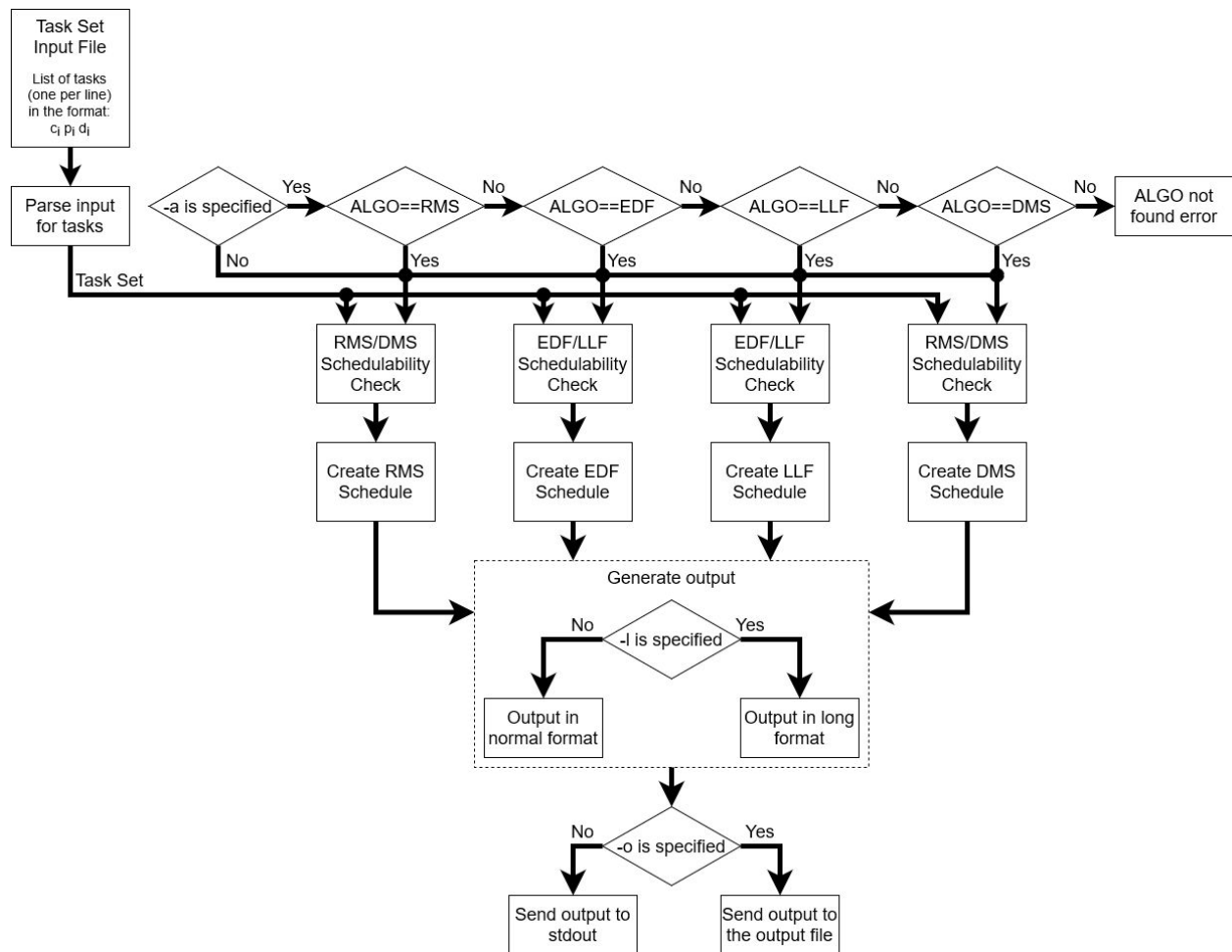
This project will involve implementation of uniprocessor scheduling through the development of a command-line utility `schedule`.

Project Goal

The objective is to build a command-line utility named `schedule` with the functionality described in the example man page (See Appendix). The main goal of this project is use what we have learned in class to develop the utility to take a text file that describes a task set and generate schedules based on the uniprocessor scheduling algorithms. Since we are a group of two, we will implement four scheduling algorithms (RMS, EDF, LLF, and DMS). The input to our command-line utility is a text file with each row representing a periodic task with three properties: c_i (computation time), p_i (period), and d_i (deadline). These properties will be represented as integers with a space between each one. The schedule will be either outputted to stdout or a user-specified file.

Solution Approach

We will implement the command-line utility in C. We will use functions from the C standard library to parse the input files and command-line options and use that to generate the output. We will write a function to check if a task set is RMS/DMS schedulable and another function to check if a task set is EDF/LLF schedulable. We will also have individual functions that will generate the schedules for each of the four scheduling algorithms. Finally, a function will output the final schedule(s) with the format depending on the options selected by the user. The system's functionality is illustrated in the figure below.



System Functionality

Expected Outcomes

The outcome will be a command-line utility that allows a user to generate a schedule (or several schedules) for a task set. The utility will have multiple options for users to select the scheduling algorithm to use, how the schedule is formatted, and where to output the schedule. The utility will be tested with multiple task sets.

References

M. Govindarasu. (2020). RMS and EDF Schedulers [PowerPoint slides].
CPR E 458/558, College of Engineering, Iowa State University,
Ames.

M. Govindarasu. (2020). RMS, EDF Schedulers (contd) -- Exact Analysis
[PowerPoint slides]. CPR E 458/558, College of Engineering, Iowa
State University, Ames.

Appendix: man page for schedule

NAME

schedule - generates a schedule for a task set

SYNOPSIS

schedule [-l] [-a=ALGO] [-o=OUTPUTFILE] INPUTFILE

DESCRIPTION

Generates a schedule for a task set.

The default operation will output a list of tasks from INPUTFILE in their scheduled order for each scheduling algorithm.

Options:

-a, --algorithm=ALGO

Schedule the task set using the specified scheduling algorithm. Possible algorithms are RMS, EDF, LLF, or DMS.

-l Outputs the schedule with the long format. Specifies the start times and end times for all tasks in the task set.

-o=OUTPUTFILE

Print the output to the specified file.

EXAMPLES

schedule TaskSet.txt

Output:

RMS: T1(0-1) | T2(1-3, 4-6)

EDF: T1(2-3) | T2(0-2, 4-6)

LLF: T1(2-3) | T2(0-2, 4-6)

DMS: T1(2-3) | T2(0-2, 4-6)

schedule -l -a=EDF TaskSet.txt

Output:

EDF: 0 T2 2 T1 3 - 4 T2 6 - 8