# ELEC 424 - Assignment 3: Rust (optional)

*The Future!?*

100 points

## Overview

You will explore general Rust coding and some of the work going on right now in the real world related to bringing Rust into kernels. The project for the latter is Theseus, which is described at that link as "a modern OS written from scratch in Rust". You will modify some of the existing Theseus code to implement some extra functionality, all in Rust. As a way to get more familiar with Rust before performing hacking in Theseus, you'll also make a blackjack game in Rust.

This assignment is optional and can replace any assignment or project grade of your choosing. You must specify the assignment number or project number in Canvas for this assignment to replace via the text box on Canvas. You can change the preferred replacement anytime by emailing me.

## Rubric

1. **(50 points) In person or submitted video demonstration (here) of functionality. Only submit one video, which can contain just blackjack or Theseus or both. In person demonstrations can be done during instructor or TA office hours.**
    a. Blackjack (25 points) - The following should be repeatable and values should be different from game to game
        i. (5 pts) Program tells user what their first two cards are and what one of the two cards of the dealer is
        ii. (5 pts) Program requests from user whether they want to hit or stand, and continues requesting until user stands
        iii. (5 pts) Program reports bust if user goes over 21 from continuing to hit
        iv. (5 pts) Once user stands, dealer shows other card (so dealer now shows two cards)
        v. (5 pts) Winner (whoever is closer to 21 but not above) is declared by program! You can handle ties however you want, including just declaring one person a winner even when there is a tie.
    b. Theseus (25 points)
        i. (9 pts) Demonstration of ls command multiplying numbers by printing result to console

        ii. (8 pts) Demonstration of ls command creating array or vector of numbers by printing array or vector to console
        iii. (8 pts) Demonstration of cat command adding numbers by printing result to console

2. **(50 points) Submission of relevant commented code files and report to Canvas**
    a. (10 points) Code attempts to achieve objectives/requirements stated in instructions
    b. (10 points) Code reasonably includes comments (at least every other line that you created/modified includes a comment)
    c. (5 points) The following file(s) submitted in their source form (e.g., .c, .dts) - not PDF
        i. .rs source file for blackjack (must include Cargo.toml file - rename to blackjacl_Cargo.toml)
        ii. Modified .rs file related to ls (must include Cargo.toml file if modified - rename to ls_Cargo.toml if included)
        iii. Modified .rs file related to cat (must include Cargo.toml file if modified - rename to cat_Cargo.toml if included)
    d. (25 points) PDF report that includes:
        i. (1 point) Title of assignment/project
        ii. (1 point) Your name
        iii. (5 points) 1 paragraph (at least 4 sentences) describing the goal of the assignment and the steps you took to complete it (include a statement on each key function)
        iv. (5 points) A 2nd paragraph (at least 4 sentences) describing what you found challenging/any bugs you had to fix, what you learned, and what you think would be another interesting application for Rust.
        v. (6 points) Include three screenshots showing a significant portion or all of your code.
            1. One screenshot for blackjack
            2. One screenshot for ls
            3. Onc screenshot for cat
        vi. (4 points) Include a screenshot of terminal output showing the messages printed by your code for blackjack and another screenshot showing some output for your ls and cat commands in Theseus.
        vii. (3 points) All screenshots in the report must include a figure label with a short description.

## Guidelines

- **Blackjack: Create a blackjack game in Rust**

    - See the rules of blackjack here if you are unfamiliar with the game
    - Implement a simple version of blackjack in Rust
    - The game/program should give the user and dealer two cards at the beginning
    - Card values will be 1 to 11
        - (aces will be low (1) or high (11) automatically - ignore this point if it confuses you, drawing 1 to 11 randomly automatically address this point)
    - The program will print out your two card values and one of the dealer's card values (the dealer's other card is a secret until later)
    - Then program will ask the user if they want to hit (get another card) or stand (draw no more)
        - The user can continue to hit, but must stop once choosing stand
        - Program immediately reports bust if user goes over 21 from continuing to hit (and game ends)
    - Once the user finally stands, the dealer's other card should be printed along with the dealer's first card (print both of the dealer's cards)
    - Whoever is closer to 21 (and not above) should be printed as the winner
        - You can handle ties however you want, including just declaring one person a winner even when there is a tie.
- **Theseus: Perform hacking of Theseus to add functionality to two commands**
    - Decide which platform  (WSL, Mac, Linux, or Linux Virtual Machine)  to run Theseus on - don't run it on your RPi (I don't believe it will work). You shouldn't be able to run it on CLEAR because of the sudo requirement.
        - Follow the Theseus installation instructions for your platform here
        - Before running Theseus, read the instructions on the repo on "Using QEMU" and make sure you understand how to quite QEMU
            - You can also pull this up on your phone
            - The issue is that QEMU will try to keep you trapped in its window! So you have to know how to quit it depending on your platform
        - Then you can run Theseus via the command "make run" (I have to run gmake run on my Apple M2 Max)
            - Once the prompt appears, test the ls command to make sure it works - you should see some folders, yay!
            - Now you can quit Theseus/QEMU
    - Now your mission: Modify Theseus' ls and cat commands
        - All of these modifications must be done in the corresponding source files
            - You should only need to modify two .rs files (and possibly the associated Cargo.toml files)
        - Modify `ls` to enable using a fake filename where the syntax of the filename is operation_number1_number2_number3… (and so on; you

must support variable length here) so that the command “ls operation_number1_number2_number3…” results in:

            - Multiplication of numbers if operation is “multiply”
                - Print result of multiplication to console
            - Array/vector creation if operation is “array” or “vector” (you only need to support arrays or vectors, not both - you can choose)
                - Print array/vector to console
            - It is fine if an error is printed after this in Theseus - that is not a concern for this assignment
            - Here are example uses of our modified ls command:
                - **`ls multiply_2_3`**
                    - Should print “6” to the console
                - **`ls multiply_2_3_6`**
                    - Should print “36” to the console
                - **`ls array_1_2_3_4`**
                    - Should print something like “[1,2,3,4]” to the console
        - Modify cat to enable using a fake filename where the syntax of the filename is add_number1_number2_number3… (and so on, you must support variable length here) so that the command “cat add_number1_number2_number3…” prints the sum of the numbers.
            - Here are example uses of our modified cat command:
                - **`cat add_2_3`**
                    - Should print “5” to the console
                - **`cat add_2_3_6`**
                    - Should print “11” to the console
    - After any modifications, you will want to run Theseus again. Then you can test the commands (that have had their associated code modified) in Theseus.