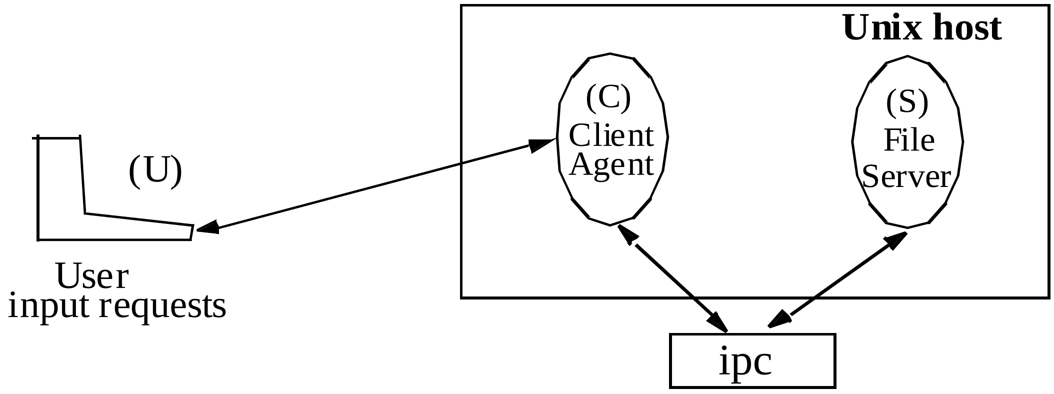
# Santa Clara University

Department of Computer Engineering
(COEN 236)

## Project-4

### Project Overview:

In this project, we would like to re-implement the file server application discussed in either projects (1 or 2) using sockets covered in the class. The client agent process reads input commands from the user (U), pass the operation required to the file server (S) through our IPC mechanism, gets the result back and pass it back to the user (U). The interactions between the user, client, and the file server are synchronous, i.e., the user has to wait to receive the result of an operation before requesting another operation, etc.



### Functional Requirements:

This project requirements are similar to project (3) with the exception of using different IPC mechanisms and that the client and server to be on different hosts. We will use sockets as discussed below:

1. Use sockets with connection-oriented protocol. Design your own message format between (**C**) and (**S**).
2. Use sockets with connectionless protocol. Design your own message format between (**C**) and (**S**).

## **Programming Hints**:

1. Structure your code to minimize reimplementation, i.e., maximize code reuse, of every module for the different IPC mechanisms.
2. Your agent process (C) should be structured into four main segments:

   - *Initialization segment*: setting up signals, opening the shared file, creating the server process, etc.....
   - *Command input segment*: read a line command from the standard input, check legal syntax, and branch to execute legitimate commands.
   - *Command execution segment*: if file access operation forward it to the server and wait for the result. If exit do cleanup and terminate.
   - *Output segment*: write the operation result on the standard output, and asks for next command.

3. Include a brief description of your project implementation for every IPC mechanism used. Remember to have good comments in your code. Also, include a hard copy of user interaction with your file server application.