There are 4 questions; answer all of them. If you need to make an assumption to clarify a problem, write your assumption down. Only reasonable assumptions get full credit. Explain all answers. You have one and a half hour to finish the exam. Good Luck!

1. Name an operating system that is not “simple and elegant”. (1 pt)

2. Interprocess Communication (33 pts.)

Consider an operating system that supports the following operations (33 pts.):

```
async_send (receiver, message, msg_len)
send_wait (receiver, message, msg_len, reply_msg, reply_len)
receive_wait (message, msg_len)
reply (reply_msg, reply_len)
```

`async_send` does not wait for the message to be received; `receive_wait` waits for a message from a sender; `send_wait` waits until the receiver replies using `reply`; and `reply` is asynchronous. The sender and receiver may be on the same or different machines.

(a) Compare the two forms of `send`, `async_send` and `send_wait`, giving their advantages and disadvantages. (15 pts.)

(b) Consider a pair of processes, S1 and R1, that execute the following code:

```
S1
    for i := 1 to N do
        async_send (R1, message, 0); receive_wait (message, msg_len)
    od
R1
    for i := 1 to N do
        receive_wait (message, msg_len)
    od
```

and another pair, S2 and R2, that execute the following code:

```
S2
    for i := 1 to N do
        send_wait (R2, message, 0, reply_message, reply_len); reply (S2, reply_message, 0)
    od
R2
    for i := 1 to N do
        receive_wait (message, msg_len)
    od
```

The following four experiments are performed and the times taken by the sending process to execute the loop observed:

1. S1 and R1 execute on the same computer. Time taken: 4 seconds.
2. S1 and R1 execute on different computers. Time taken: 2 seconds.
4. S2 and R2 execute on different computers. Time taken: 7 seconds.

Explain why experiment (2) takes less time than (1), (3) takes more time than (1), and (4) takes more time than (3). If you cannot justify this difference, explain why not. (18 pts.)
3. Process Management and Coordination: (33 pts.)

(a) Outline an implementation of the Xinu suspend() operation. You can ignore the fact that this operation returns a value. (8 pts.)

(b) Consider a system that (unlike Xinu) supports both monitors and the suspend() operation. What should be the semantics of invoking the suspend() operation on a process executing a monitor entry procedure? If you can think of multiple possible semantics, give all of them, and discuss their relative advantages and disadvantages. Identify the semantics you prefer and explain your choice, using at least one example. (25 pts.)

4. Terminal Driver: (33 pts.)

(a) Give and justify the rules to be followed while writing interrupt routines. (17 pts.)

(b) Suppose we wished to embellish Xinu with an interactive facility to suspend and resume processes: When the user enters a special "suspend character", the current process is suspended as long as it is not the null process. (The current process is the process that executes the input interrupt routine called in response to the entry of the suspend character.) When the user enters a special "resume character", the last (unresumed) process suspended by the user is resumed. Can such a facility be added to Xinu? If your answer is no, give reasons why it cannot be added. If your answer is yes, outline an implementation of the facility, and give revisions to the interrupt rules of part a) if necessary. (16 pts.)