

# COEN 10

## Lab 7

# Lab 7 – Scheduling System

- ❖ Your program will deal with a dentist appointment system.
- ❖ The dentist is available each day from 1pm to 7pm for 1-hour appointments.
- ❖ Your system creates appointments for one day only.
- ❖ The system reserves the earliest appointment available.

# Lab 7 – Scheduling System

## ❖Interface

◆The user can use the system to

- Schedule (1) an appointment
- Cancel (2) an appointment
- List (3) the appointments
- Quit (any other number)

# Lab 7 – Scheduling System

## ❖Interface

### ◆Create an appointment – enter name

- If there is a free hour
  - The user is asked for his/her name
  - The appointment is scheduled in the earliest free time slot. The hour is shown to the user.

### ◆Cancel an appointment – enter name

- If there is a room reserved
  - The user is asked for his/her name
  - If found, the appointment is canceled and later appointments are shifted up

### ◆List appointments

- Show the schedule, all the names and free slots

### ◆Quit

- Return from the main function

# Lab 7 – Scheduling System

## ❖ Implementation

- ◆ Use an array of strings, size 6x20
  - 6 appointments
- ◆ Initially, the array contains a null character ('\\0') in the first position of each string
- ◆ Keep a counter of number of appointments

# Lab 7 – Scheduling System

## ❖ Implementation

### ◆ Schedule

- If the dentist is busy, inform the user
- Otherwise
  - An appointment is scheduled in the first slot available, given by the number of appointments (no loop)
  - Update the number of appointments

# Lab 7 – Scheduling System

## ❖ Implementation

### ◆ Cancellation

- If the schedule is empty, inform the user
- Otherwise, search for the name in the array
  - If found
    - » Cancel the corresponding appointment
    - » Shift later appointments up to close the hole
    - » Update the number of appointments
  - If not found
    - » tell the user

# Lab 7 – Scheduling System

## ❖ Implementation

### ◆ List

- If the schedule is empty, inform the user
- Otherwise
  - Traverse the array, showing the name assigned to each appointment or an empty string for the free slots.

# Lab 7 – Scheduling System

## ❖ Requirement

### ◆ Have a forever loop

- In the loop, use switch to decide which action to take depending on the command entered: 1, 2, 3, or any other

### ◆ Variables

- array of strings to keep the appointments
- number of appointments

# Lab 7 – Scheduling System

❖ You will use C in the Mac or Linux

◆ Use your DC account

- The home directory
- You don't need to do this on the web server

◆ Edit the program using vi in the terminal

- The program needs to be a “.c” file

◆ Compile with gcc

`gcc -o name name.c`

◆ Execute

`./name`

# Lab 7 – Scheduling System

## ❖ When you are done

### ◆ Demo

- Execute your code on the terminal to the TA

### ◆ Submit

- Print and submit the source code to the TA
- Don't forget to put your name on it!