# Laboratory 6

# CST 8244

# Professor: Dr. Douglas King

# Due: Nov 10, 2016

## Instructions

You will be writing a program that operates a beat-box using timers. (For the beat – just send a bell character to the display).

You will be creating 2 programs:

1. **beatDevice devicename &**

Installs a device driver (named according to the command-line parameter: e.g., /dev/local/beats/devicename) that makes sounds (in our case just using the bell character) when string data is written to the device. The device must be designed to let multiple programs connect to the device so that the beats can overlap. The beatDevice should respond to the following values being written to the device

* 1. **if “1” -> single bell, then pause of 50 msec**
  2. **if “2” -> two bells, then pause of 50 msec**
  3. **if “3” -> two bells, then pause of 200 msec, then another bell, then a gap of 50 msec.**

1. (interactive) **beatInput devicename speed**or (non-interactive – using the “-loop” command-line argument)   
   **beatInput –loop filename devicename speed &**

If started in interactive mode, prompts for input strings (separated by newline characters) from the console and writes them to the named beatdevice. The **speed** parameter is the integer number of milliseconds to delay between beats. (0 is hyper-fast; max value is 5000).

* 1. **if b1 is entered, write -> “ 1 “**
  2. **if b2 is entered, write -> “ 2 “**
  3. **if b3 is entered, write -> “ 3 “**
  4. **if d### is entered, delay for an extra ### number of milliseconds before sending the next beat.**

If the beatInput is started in loop mode, then the input strings (separated by newline characters) should be read from the named file instead of the console, and the input should loop (i.e., start again at the beginning of the file after reaching the end).

**Note**: The important thing is to have the ability to run multiple devices and multiple inputs at the same time. Be prepared for this to be tested during the lab demo in a variety of combinations.

## Marking Scheme

This Assignment will be out of 60 marks. (notice the error-checking….)

* 10 marks for proper use of timers
* 30 marks for proper use of device driver(s) & multiple threads.
* 10 marks for the proper output
* **10 marks for error checking.**

Marks will be deducted for not checking return codes for errors or not properly freeing any dynamic memory you allocate. Comments are not necessary but they are always welcome, especially if you do not fully complete the programs and want to receive partial marks for what you have completed.

## Submission Guideline

You will demo the lab to the professor during the laboratory period, and then submit your project directory to Blackboard.