The University of Queensland
School of Information Technology and Electrical Engineering
Semester Two, 2017
COMP3301 / COMP7308 – Assignment Three
Due for demonstrations: your prac session - week 13 (25-26th of October), 2017
Weighting: 30% of your overall mark (COMP3301/COMP7308)

Assignment Three – Remote Controlled Webcam Display

Objectives:

- To apply your knowledge and skills developed in pracs 1 to 6.
- Use IO Driver concepts.
- Use advanced inter-process sharing techniques.
- Apply memory management concepts.

Introduction:

This assignment will build on your knowledge from pracs 1 to 6. You will be working individually on all tasks. This assignment asks you to develop a remote control webcam display. The focus of this assignment is using IO interfaces and advanced inter-process sharing techniques.

Course Pass Hurdle:

To pass COMP3301/7308, you must achieve 50% or more of the marks for Assignment 3. If you do not pass assignment 3 (49% or less), then your final course grade will be capped at level 3.

Requirements:

NOTE: Version Control – You MUST use version control as part of your development process or you will not be assessed. This means that because the assignment is a bigger task than the pracs, you should have a NUMBER of commits including the FINAL version which is the one being assessed. If that is not the case, you will not be assessed. You should put your code in an assignment3 folder on git.

You must follow the style guide, listed on Black Board[4].

NOTE: YOU MUST BE MARKED IN YOUR ASSIGNED SESSION. You will **NOT** be marked if you attend another session.

Due Dates:

The **assignment** is due in the prac session, in **week 13.** The assignment includes Tasks 1-4. Task 5 (challenges – must only be attempted if Tasks 1 to 4 are complete). In week 12, non-compulsory attendance help sessions will be available.

IMPORTANT: This assignment **MUST** be implemented the Raspberry Pi 3 platform running Raspberian Pixel OS.

Design Task

Develop a webcam display (WCD) using the RPI3, webcam and remote control. The WCD should be able to display four types of imaged processed outputs captured from the webcam. You will need to use the remote control to control each processed output. Figure 1 shows an overview of the WCD system.

The WCD system consists of the following separate programs:

- ***WCD Shell*:** Controls the WCD System and provides a user shell interface.
- ***WCD Display*:** Displays a webcam frame in a corner of the screen.
- ***WCD Remote*:** Processes commands received from the remote control
- ***WCD Camera:*** Control the webcam.
- ***Memcached:*** Sharing common variables between programs.

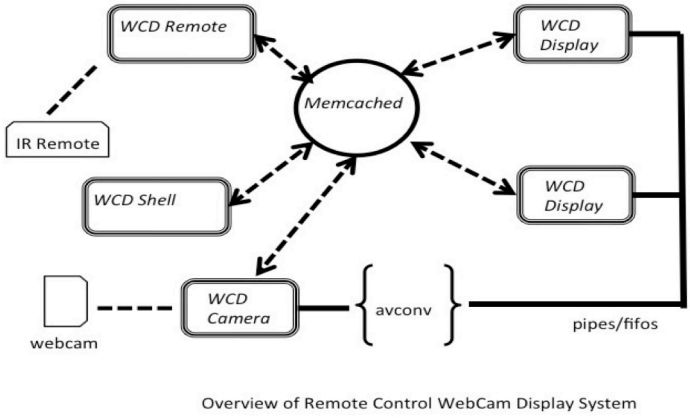

Figure 1: Remote Controlled Webcam Display Overview

(3 marks) Task 1: WCD SHELL

Table 1 lists the user shell commands to be implemented. Extra commands can be added but will not be assessed. A prompt should be displayed. The user shell should display a command prompt in red: ":^) ". The command text must be in displayed in green when typed. Memcached should be used to share any parameters with the ***WCD Display***. Also see Table 3 for commands implemented by the ***WCD remote***. Note: other shell commands for testing purposes can be added.

Table 1: User Shell Commands

Command	Parameter	Description
wcd	<x, y, width, height>	Create a WCD display at the x, y coordinates, with a certain width/height (pixel). See Task 4.
sys		See Task 4.
clear	<WCD ID>	Clear the WCD display. See Task 4.
list		Show the WCD ID values and locations. See Task 4
kill	<WCD ID>	Remove a WCD display. See Task 4.
sel	<WCD ID>	Select a WCD display. See Task 4.
img	<command>	Use avconv to process the video input and display the output (e.g. transform, blur, black/white, slow rate) <command> can be any number of parameters that's required.

(7 marks) Task 2: WCD Webcam Control

The WCD webcam control program configures and captures frames from the webcam. The WCD webcam control program must use the video4linux2 libraries. Once an image has been captured, it can be processed and displayed. Pipes/FIFOs should be used to transfer the image stream to other programs. To process the image, the `avconv` program (using pipe interface) should be used. Below are the following processed image functions that that should be displayed. The ***WCD Remote Control*** program and the `img` shell command can be used to select which process image function to do. The basic functions that should demoed, are listed below

a) Raw (not processed) - (Remote enter – 'RAW' (no parameters))
b) Flipped (transformed) – (Remote enter – FLP (to 180 degrees))
c) Delayed (down-sampled) (Remote enter – DLY (specify rate))
d) Blurred – (Remote enter – BLR (no parameters))

(5 marks) Task 3: WCD Display

The ***WCD Display*** program displays a webcam frame ***stream*** *(continuous images – FPS should atleast be 4)* in at a particular location. The shell command `wcd` should be able to create a new instance of the WCD program. You will need to be able to create two ***WCD Displays***. Likewise, the command `kill` should be able to remove an instance of the WCD Display program. Each WCD Display program should The ***WCD Display*** program should take as command line arguments the x, y coordinates and the width/height (pixels) of the display. The x and y coordinates position the bottom left corner of the ***WCD Display*** on the screen. The ***WCD Display*** can be implemented with X11 or with other graphical libraries (e.g. opengl, openvg, etc – must be approved before using). Each instance of the ***WCD Display*** should a unique ID (single digit) that should be stored in Memcached and listed using `list`. Note the WCD ID will be used by the WCD remote control, so it should be a single digit. Selecting which WCD to display the latest captured image, is set with the ***WCD Remote Control*** program. Memcached must be used to retrieve the key values from the ***WCD Remote Control***.

(5 marks) Task 4: WCD Remote Control

Use the Infrared remote to control the WCD. The remote should be used to input alphanumeric text, that will be displayed on the WCD Shell. The numeric keys on the WCD should implement a telephone keypad for alphanumeric input[5]. Pressing the same numeric key should cycle through the options. E.g. pressing '2' three times should select 'B'. Table 2 shows the remote keypad assignments for the WCD.

Table 2: WCD Remote Control Keypad Assignment

Keys	Key function
0	0
1	1
2	2, A, B, C
3	3, D, E, F
4	4, G, H, I
5	5, J, K, L
6	6, M, N, O

7	7, P, Q, R, S
8	8, T, U, V
9	9, W, X, Y, Z
Arrows or VOL/CH	Move selected image to left/right/up/down to the next WCD Display.
SETUP or RECORD	Select the current WCD display.
Time Shift or STOP	Clear selected WCD Display.
Mute or Vol+	Clear all WCD Displays.

The ***WCD remote*** is a separate program that must use the LIRC driver to receive key presses from the remote. Memcached should be used to store and share the received remote keys between all programs. You can either write the ***WCD remote*** to either use the LIRC Daemon or to function as a low level driver (inserted using modprobe). Memcached must be used to pass the key values to the ***WCD Display***. Table 3 shows the commands that the must be implemented by the ***WCD remote***.

Table 3: WCD Remote Control Commands

COMMAND	**Key function**
S	Start display capture streaming
RAW	Display raw image
FLP	Flipped (transformed) – (Remote enter – FLP (to 180 degrees)
DLY <rate>	Delayed (downsampled) (Remote enter – DLY (specify rate)
BLR	Blur Image

Total: 20 marks (20% of course marks)

(10 marks) Task 5: Challenges (Must only be attempted, if Tasks 1 to 4 have been completed)

The follow tasks are challenges, which add advanced features to the WCD. You must complete Tasks 1 to 4, **BEFORE** attempting the challenges. You can chose to do a combination of challenges but you cannot do all the challenges. You cannot get more than 10 marks.

1) Turn your RPI3 platform into a standalone WCD by being able to plug the RPI3 directly into a HDMI capable monitor. Use a headless version of Raspberrian and with the openVG library for the graphical output. Create WCD services to automatically run your WCD system, on boot up. The entire system should be self contained and not require an user input or action to initialise on start up. Commands should only be issued using the remote control. RPI3 must only be plugged into a HDMI monitor (you must supply the HDMI cable). **- 10 marks**

2) Implement the other image transforms using avconf, such as imposing text on the image stream. **- 5 marks**

3) Display realtime graphs of the process, memory usage statistics from the /proc folder, using X11. **- 5 marks**

4) Implement the `sys` command to show the LIRC kernel module usage. You must use the /proc folder to extract the information to be displayed. **– 5 marks**

For the LIRC kernel module usage, list the following:

- The memory size of the module, in bytes.
- How many instances of the module are currently loaded.
- What load state the module is in: Live, Loading, or Unloading.
- Current kernel memory offset for the loaded module.

5) Suggest your own challenge. This must be approved by the course coordinator before you attempt it. **– 5 or 10 marks**

Total: 10 marks (10% of course marks)

Mark Deductions

All code must function and be able to be compiled, in order to receive any marks. No marks are given for non-functional or non-compiled code.

5 mark deduction) – Incorrect use of threads and processes

5 mark deduction) – Specified libraries/programs were not used.

5 mark deduction) – Incorrect use of inter-process communication.

2 mark deduction) – Style Guide is not followed.

2 mark deduction) – Program naming convention (e.g. sxxxxxx_program name).

All code used to demo your assignment and **must be your own work.** You are NOT permitted to use other code sources, unless directed to or previously approved. Plagiarism is unacceptable. You were required (course profile of this subject) to read and understand the School Statement on Misconduct, available on the ITEE website at: http://www.itee.uq.edu.au/itee-student-misconduct-including-plagiarism .

References

1. Memcached - https://memcached.org/
2. C Style Reference Guide – See Black Board\Guides
3. X11 reference - https://www.x.org/releases/X11R7.7/doc/libX11/libX11/libX11.html
4. COMP3301 Lectures 9, 10 and 11 on Black Board.
5. https://en.wikipedia.org/wiki/Telephone_keypad
6. avconv - https://libav.org/avconv.html