



Week 06

Students are expected to attempt ALL of the questions in the Multi choice, Short answer and Lab questions. They will be discussed and marked at the beginning of the lab.

Multi choice

BC

B

Question: Which of the following statements is false?

- 1) Every variable in C has an address regardless of its type (except variables that are just stored in registers - which is really more of a compiler optimization)
- 2) For a variable defined as: `int x;` you can modify the its address by saying `&x = 1234;`
- 3) On a 64-bit processor the statement: `sizeof(int*)` would resolve to the number 8
- 4) The statement `*x = 0;` is valid for a variable defined: `int x[5];`

- A) 1
- B) 2
- C) 3
- D) 4
- E) None of the above

A

Question: How do you open a file called "data.txt" such that it is write only, created if needed, and truncated to an empty file?

- A) `int fd = open("data.txt", O_CREAT | O_WRONLY | O_TRUNC, 00666);`
- B) `int fd = open("data.txt", O_WRONLY | O_RDONLY, 00666);`
- C) `int fd = open("data.txt", O_WRONLY, 00666);`
- D) `int fd = open("data.txt", "w+");`
- E) `int fd = open("data.txt", O_CREAT & O_WRONLY & O_TRUNC, 00666);`

B

Question: Which of the following statements are true about a struct?

- 1) A struct can store `void` typed variables
- 2) Calling `sizeof()` on a struct won't necessarily return the exact sum of bytes used by variables within the struct
- 3) Structs can contain function definitions

- A) 1 only
- B) 2 only
- C) 3 only
- D) 1 and 3

size of only return the number of bytes.
 if one struct occupies 31 bits, take `sizeof()` of that struct
 will return a integer that denotes # of bytes. that is,
 denoting a multiple of 8 bits - a result is not exact
 number of size of that struct.

Short answer

Question: Explain what the following statements mean. Also say if any variables should be pointers for the statement to be legal.

1. `a * b`
2. `*a = b`
3. `a = &b`
4. `int b[5]; a = *(b + 4)`

1. variable a times variable b
2. assign value b to the address that a points to
3. assign the address of b to variable a
4. state b to be a array of interger with 5 elements.
assign 5th element of array b to variable a.

TBD

Question: In a Unix-based operating systems, how does an application program interact with files? What are some advantages and disadvantages in the approach taken by unix systems? fopen with many flags and modes

Question: What are the two different ways to get an element from a struct (assuming you have a pointer to the struct)?

1. use corresponding pointer to access that struct. say, `pointer -> member`
2. use corresponding structure variable to access that structure. that is, `var.member`

Lab questions

Exercise: Write a program which opens a file which has a list of names and people's heights and generates a report to standard output. The input file format contains names and heights on each line and the name and height is comma seperated (assume the name length is atmost 30 characters). You should make use of an array of "struct" to store all the peoples details (you may assume there is atmost 100 names/heights in the file). The report should include the average height and the name and height of the tallest and shortest people. The file name can be fixed to "data.txt". A typical interaction with the program would be:

```
% cat data.txt
Eric,5.7
Bill,4.8
Jill,5.9
Mill,5.5
% ./reportheight
```

DONE, see reportheight.c file

```
Of the 4 people the average height is 5.5m, the tallest person is Jill at 5.9m, and shortest is Bill at 4.8m.
```

In-class group task

The goal of this group exercise is to gain some more practice and experince in programming c. To this end you are required to write a program that controls a car on a race track. The cars will race against each other over the network. However, to start with you will get your car going by itself. This excercise may be completed in groups of 2 or 3 people.

Write a client:

1. Download cargame.zip

2. In the file `client/control.c`, implement the logic to accelerate and steer a car based on input received from the five sensors you can see in the image below.
3. Compile it by opening a terminal in the client folder and typing `make`. Do this each time you make a change.

Test the client:

1. Open a terminal in the server folder and type `java -jar server.jar track1.png 12345` (**Note:** The track and port number can be different)
2. Open another terminal in the client folder and type `./client localhost 12345`
3. Press `Ctrl+C` to stop the client

If you look in the `client.h` file you will find all the input and output values for your car:

InputPacket:

- `acceleration`: How fast the car is accelerating right now in pixels/second/second (max: 100)
- `steering`: How fast the car is turning right now in radians/second (max: 5)
- `speed`: How fast the car is going right now in pixels/second (max: 100)
- `laps`: The number of laps the car has done so far
- `sen_mid_dist/sen_mid_type`: The distance the middle sensor is detecting, and what type of surface it is detecting
- `sen_close_left_dist/sen_close_left_type`: The distance the closer left sensor is detecting, and what type of surface it is detecting
- `sen_close_right_dist/sen_close_right_type`: The distance the closer right sensor is detecting, and what type of surface it is detecting
- `sen_left_dist/sen_left_type`: The distance the far left sensor is detecting, and what type of surface it is detecting
- `sen_right_dist/sen_right_type`: The distance the far right sensor is detecting, and what type of surface it is detecting

Each sensor is 30 pixels long.

These are the surface types:

- `SEN_NONE`: Not detecting anything, just road
- `SEN_WALL`: Detecting a green wall
- `SEN_CAR`: Detecting another car. The sensor line will turn yellow
- `SEN_FINISH`: Detecting a red finish line

OutputPacket:

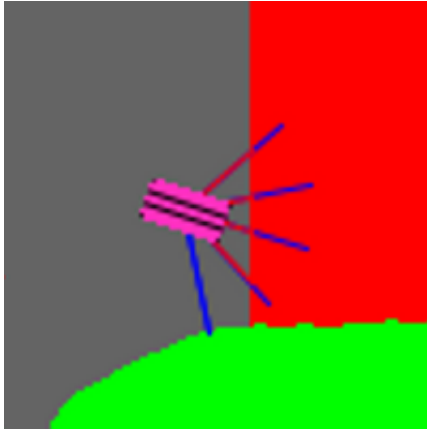
- `acceleration`: What you want the acceleration to be in pixels/second/second (max: 100)
- `steering`: What you want the turning speed to be in radians/second (max: 5)
- `tint_r`: What you want the red tint of the car to be (in the range 0-255)

- `tintg`: What you want the green tint of the car to be (in the range 0-255)
- `tintb`: What you want the blue tint of the car to be (in the range 0-255)

Use `tintr/tintg/tingb` to change your car's colour and make it unique from everyone else's car.

At the end of the lesson you will be able to connect to the tutor's server and race your car against others.

Hint: Look at `client.h` for the possible input/output values.



Press 'r' to reset all the cars back to the start line.

Once you have complete a few races your tutor will get you together and as a group you can share the different approaches groups took in implementing their car.

UPDATED: 12 March 2013 / RESPONSIBLE OFFICER: Head of School / PAGE CONTACT: COMP2300 Course Webmaster