COMP5514
Computer Image Generation in C/C++

Lab 01

Prof. George Baciu
csgeorge@comp.polyu.edu.hk
www.comp.polyu.edu.hk/~csgeorge/comp407/lab/

Department of Computing, --- The Hong Kong Polytechnic University

Contents:

Part A: CygWin

- CygWin in the Lab
- CygWin @ Home
- CygWin Installation
- CygWin User Setup
- CygWin Navigation

Part B: make g++

- MingW32-Make
- Make Utility
- Makefile
- Setting up your 1st C++
- Compile, Link, Run

Part C: Hello C++

- Basic C/C++ program
- Compile, Link, Run
- Makefile

Part A: CygWin – Linux in a Window

- **CygWin: Unix/Linux in a window!**
  - No need for dual boot
  - Can access all files – bidirectional: CygWin ⟺ Windows
  - Very small kernel
  - Portable lab – all Linux/Unix programs run
  - Very cheap: FREE!
  - Lots of tools and libraries
- **See: CygWin Manual**
- **See: CygWin User Guide and Setup**
- **Two-phase installation and … <Go>**

Part A: CygWin in The Lab

- **CygWin 1.7.7 installed in all labs under:**

  Y:\Win32\cygwin\cygwin

  - Log into your account
  - Start CygWin by running: “cygwin.bat”
  - Your home directory: “**J:\~\cyghome**”
  - Your shell environment: .bashrc .bash_profile .inputrc
  - Directory created: “C:\temp” for cygwin tmp dir

- **Mounting new mount points: cygwin.bat**

```
echo off
c:\cygwin\bin\mount -f -s -b "c:/cygwin/bin“  "/usr/bin"
c:\cygwin\bin\mount -f -s -b "c:/cygwin/lib"   "/usr/lib"
c:\cygwin\bin\mount -f -s -b "c:/cygwin/cygwin"  "/"
c:\cygwin\bin\mount -s -b --change-cygdrive-prefix
"/cygdrive"
C:
chdir C:\cygwin\bin
bash --login -i
```

Part A: CygWin @ Home

- **Download CygWin 1.7.9-1 from the web page:**

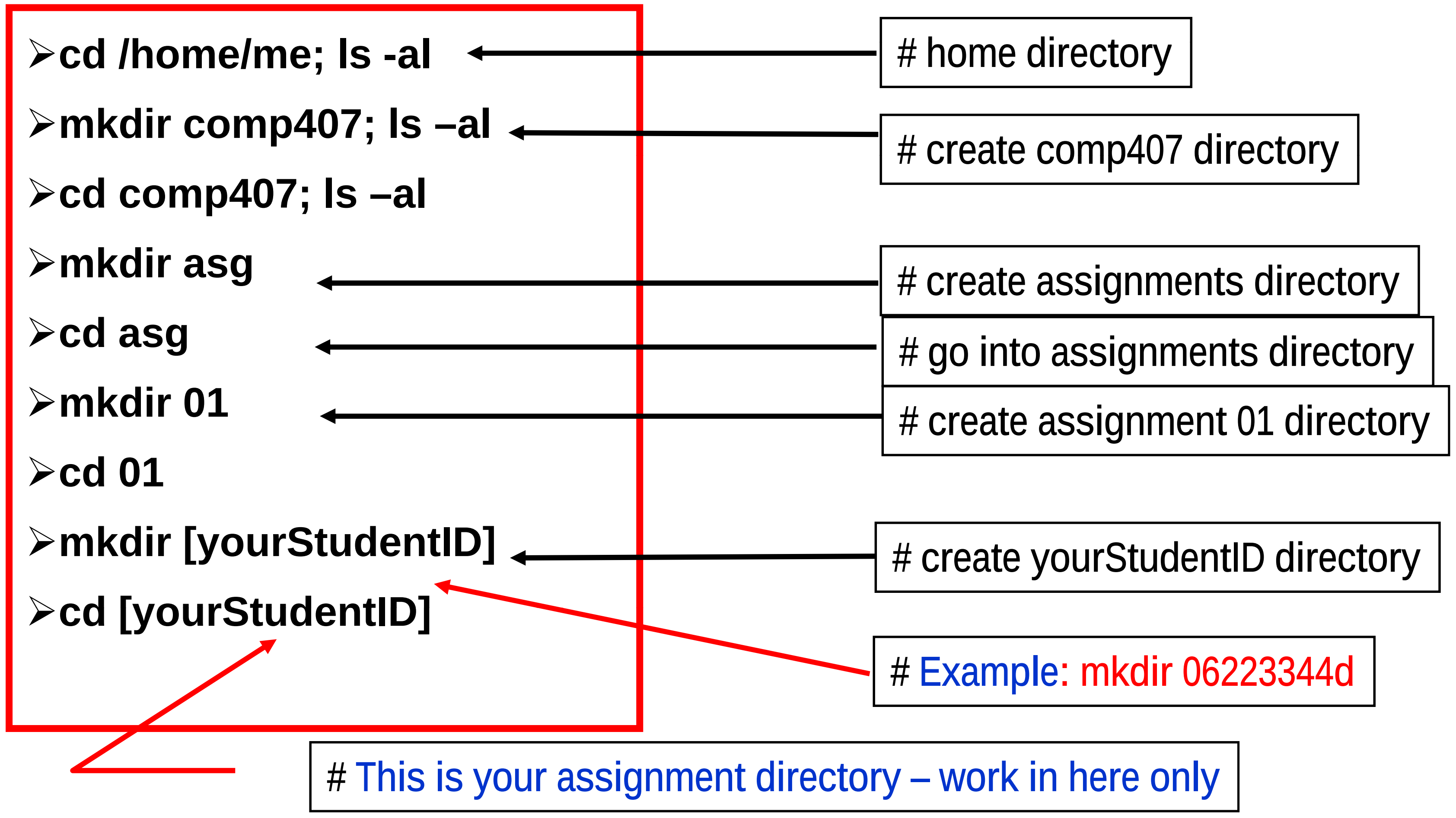
  www.cygwin.com

  - Run the cygwin setup: “Next->Next->Next->Finish”
  - Default installation: “**C:\cygwin**”
  - Make your home directory: “**C:\cygwing\home\me**”
  - Go to “/home” by: > cd /home
  - Copy Administrator to “me”: > cp –r Administrator <me>
    - Or download the user setup and copy it into directory “**me”**
  - Your shell environment: .bashrc .bash_profile .inputrc

- **Create a working directory structure – see next:**

Part A: CygWin Directory Structure

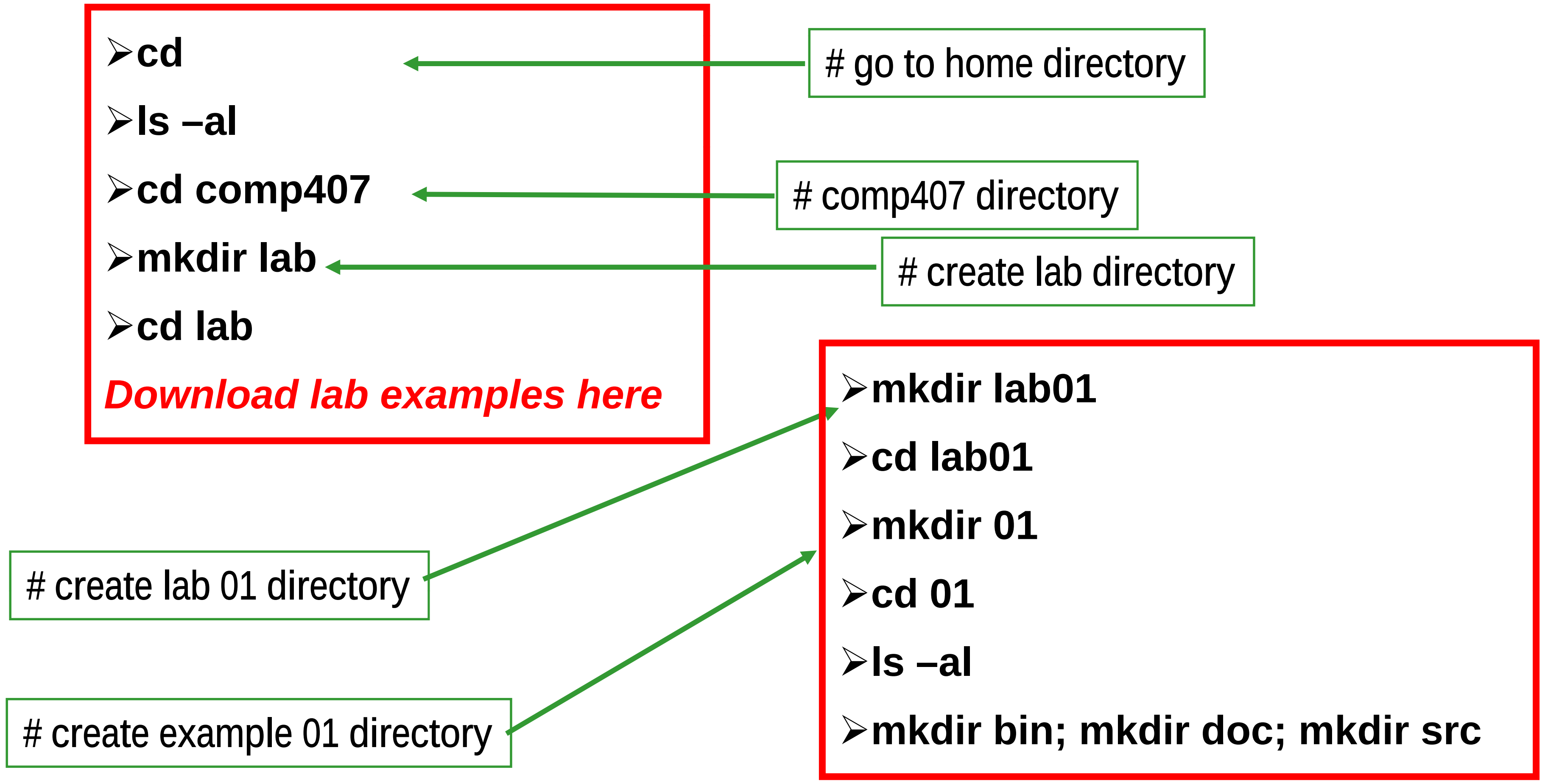
- Start up the **CygWin window**
- Create the following folders or directories:



```
➢cd /home/me; ls -al              # home directory
➢mkdir comp407; ls –al            # create comp407 directory
➢cd comp407; ls –al
➢mkdir asg                        # create assignments directory
➢cd asg                           # go into assignments directory
➢mkdir 01                         # create assignment 01 directory
➢cd 01
➢mkdir [yourStudentID]            # create yourStudentID directory
➢cd [yourStudentID]               # Example: mkdir 06223344d
                                  # This is your assignment directory – work in here only
```

Part A: CygWin Lab Directory Structure

- Start up the CygWin window
- Create the following folders or directories:



- cd
- ls –al
- cd comp407
- mkdir lab
- cd lab

Download lab examples here

# go to home directory

# comp407 directory

# create lab directory

- mkdir lab01
- cd lab01
- mkdir 01
- cd 01
- ls –al
- mkdir bin; mkdir doc; mkdir src

# create lab 01 directory

# create example 01 directory

Part C: Working with C++

- Your first program: Helloooo!
- Let's go to:

  http://www.cplusplus.com/doc/tutorial/

```
// File: main.c

// This is my first program in C++

#include <iostream>

using namespace std;

int main () { cout << "Hellooo World!"; return 0; }
```

Part C: The C part of C++

- We’ll keep things simple!

```
// Lab: comp407, Lab 01, Example 01
// File: main.c
// Description:  my first program in C
#include <stdio>
#define PI 3.14
int main (int argc, char** argv)
{
        double myPi;
        myPi = PI;
        printf(“%s\n”, "Hellooo; Anybody out there!“);
        printf(“Pi = %f\n”, myPi);
        return 0;
}
```

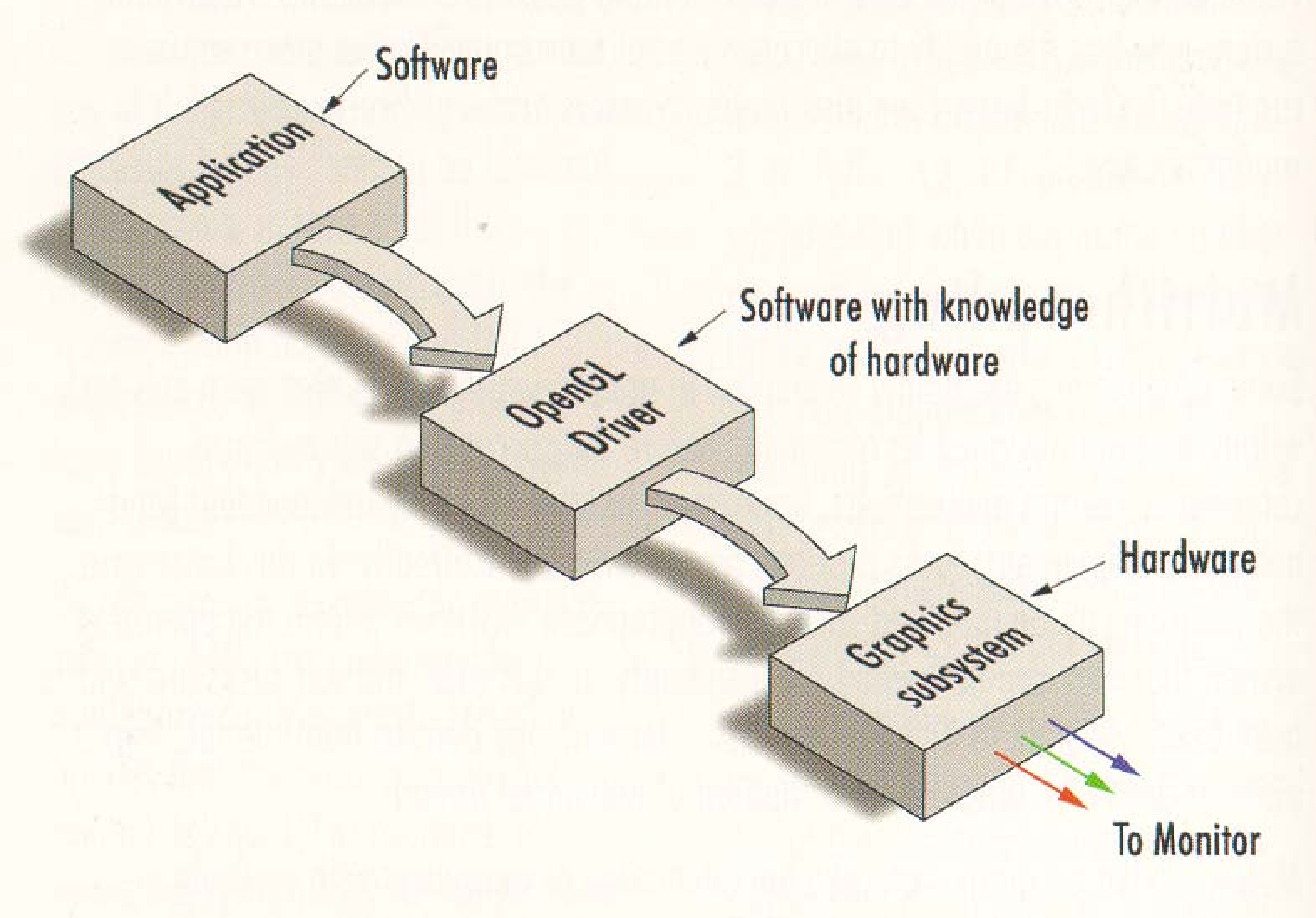
OpenGL API

- ❑Computing Environment
- ❑OpenGL Architecture

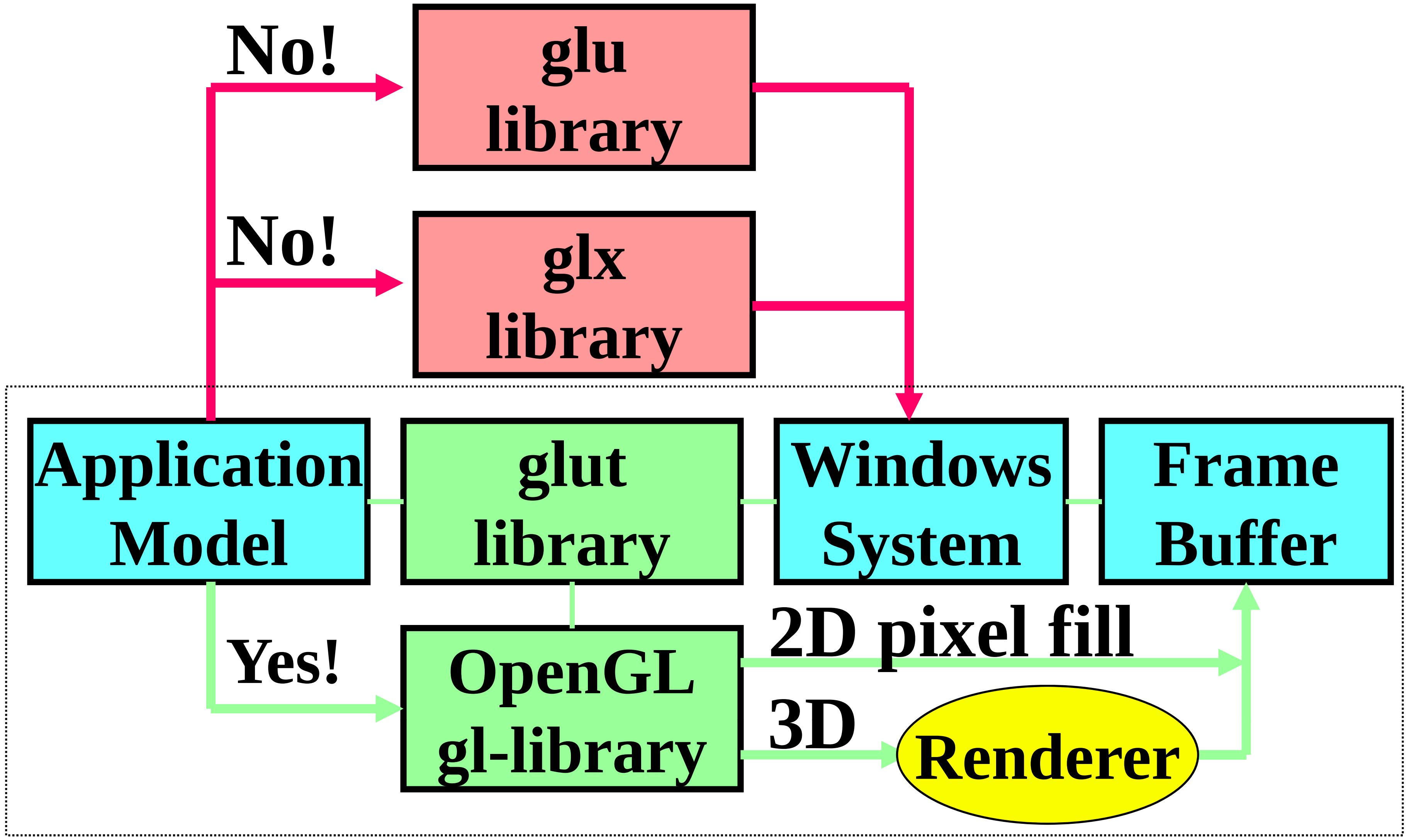
Lab Environment

- **Generic PC's**
- **Software:**
  - Windows/Cygwin
  - Gnu C/C++
  - OpenGL Library
  - GLUT Library

The Graphics API



OpenGL API


No!
glu library
No!
glx library
Application Model
glut library
Windows System
Frame Buffer
Yes!
OpenGL gl-library
2D pixel fill
3D
Renderer

OpenGL API

Library	*Prefix*	*Example*
OpenGL	gl	glColor
GLUT	glut	glutInit
GL Utility	glu	gluSphere
Auxiliary	aux	auxInitWindow
GL (sgi)	None	winopen

Paths and Files

- **OpenGL and GLUT libraries:**
  - see the link on the web page

- **Example Code:**
  - see the link on the web page

- **Map Network Drive: J:\cyghome**

Paths and Files

- **Cygwin environment**
- **Gnu C/C++:**
  - J:\cyghome

- **Use MAKE:**
  - see the example: `lab01.zip`

Interactions

- Interactions via an ***event-handler***
- Input devices generate events
- Applications must handle them
- A response is generated:
    - a visual change
    - another action

Event Handling

1. Polling: synchronous

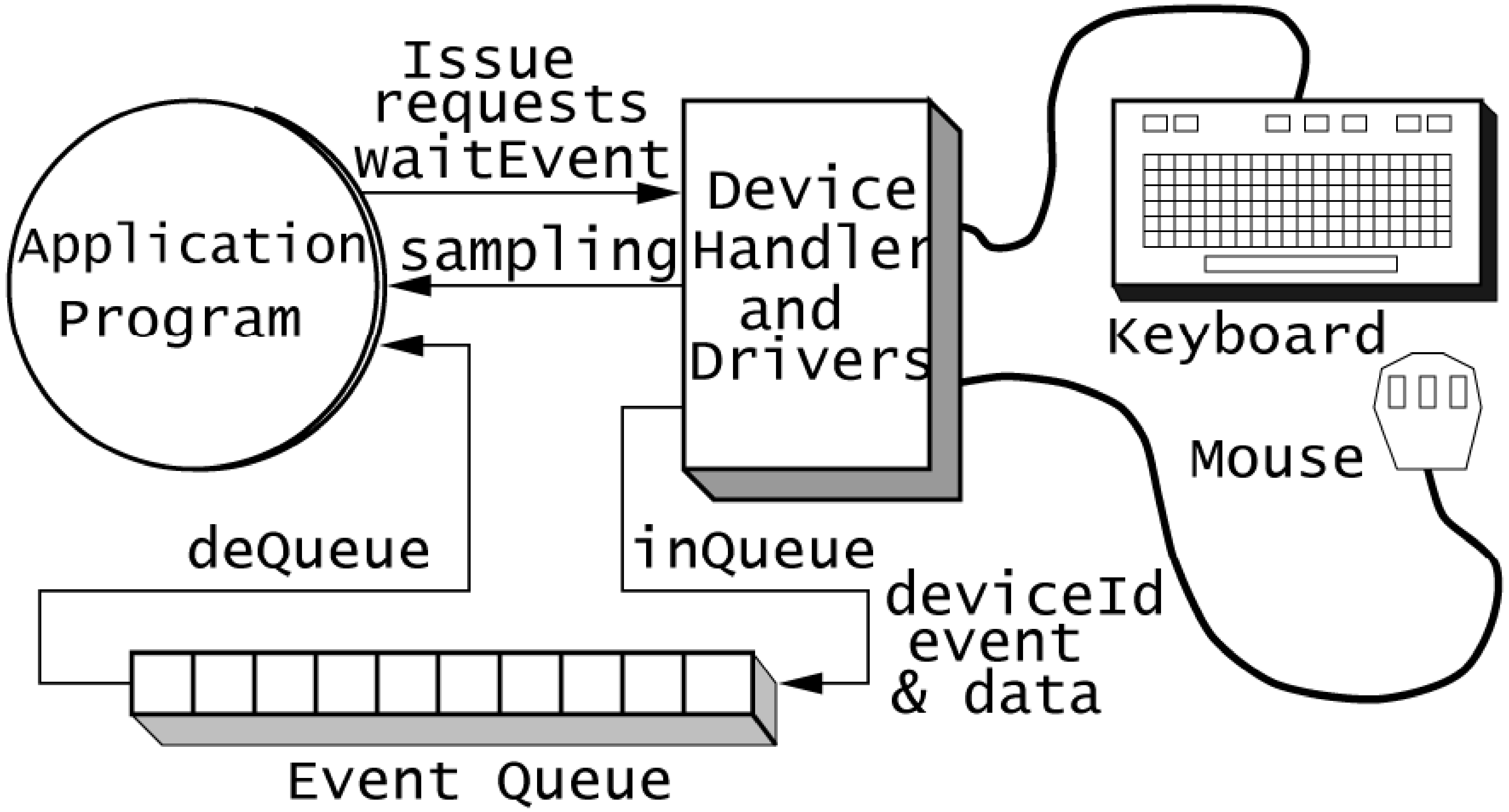
   the application is busy waiting until it receives an event.

2. Event-Driven: asynchronous

   the events generate interrupts

   events are stored in a queue

Event Handler



OpenGL Call-Back

```
int  main(int argc,  char** argv)
{       glutInit(&argc, argv);
        glutInitWindowPosition(0,0);
        glutInitWindowSize(400,400);
        glutCreateWindow(“myWindow”);
        glutReshapeFunc(myReshape);
        glutMouseFunc(myMouseCoord);
        glutMainLoop(myDisplay);
}
```

OpenGL Call-Back

```
void myDisplay (void)
{       glClearColor (0.0, 0.0, 0.0, 0.0);
        glClear (GL_COLOR_BUFFER_BIT);
        glutSwapBuffers ();
}
```

Include Files

For the Microsoft Windows environment:

```
#include <stdio.h>
#include <stdlib.h>

#include <windows.h>
#include <gl/gl.h>
#include <gl/glut.h>
```

Equivalent Loop

```
notQuit = TRUE;
while (notQuit)
{       processTheQueue();
        myDisplay();
        processTheQueue();
}
```

Queue Processing:

```
void  processTheQueue(void)
{         while ( Qtest() )
          {  device = Qread( &deviceVal );
            switch (device)
            {       case GLUT_LEFTBUTTON:
                              if (deviceValue==GLUT_MOUSEDOWN)
                                        myMouseCoord(eventDataPtr);
                    break;
                    case GLUT_ESCAPE:
                              notQuit = FALSE;
                    break;
            }
          }
}
```

Try things out. . .

The End