Agenda and Today’s Goal

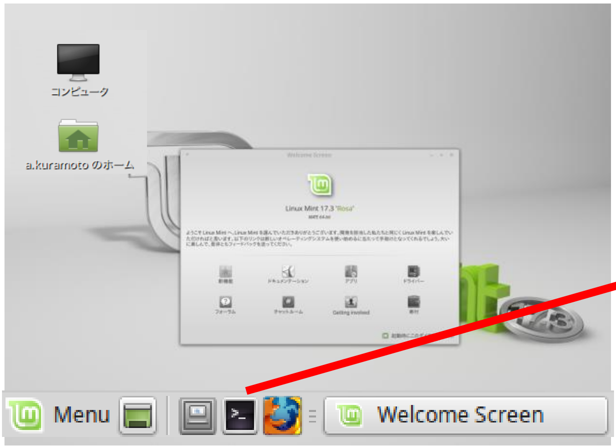
G1. Understanding Concept of File System of Linux
- ･Files and Directory
- ･Strage Device and File System
- ･Check the concept with yourself

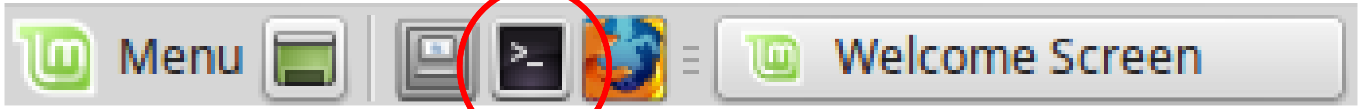
G2. Excersice the GLUT Programming using
“structure data” and “timer interrupt”
- ･ clock1.c･･･Sample program for displaying current time on terminal
- ･textDips.c･･･Sample program for displaying strings on GLUT window
- ･rectDraw.c･･･Sample program for displaying rectangle or square

【Problem】(System requirements)
- ･Display current time on GLUT window
- ･Use function “glutTimerFunc()” making cyclic interrupt

Preparing Computer system

S1:Power on HP PC , starts “Linuxmint OS” (default)
S2:Key in ID and Password
S3:Open terminal (Click here)





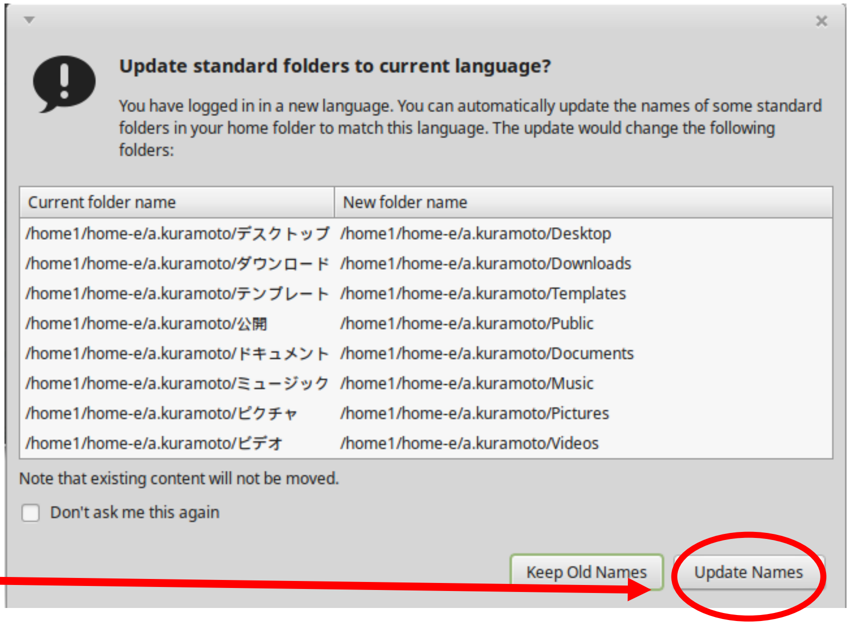
```
$ ll ↓
```

S4:Change English mode

```
$ LANG=C xdg-user-dirs-gtk-update ↓
```

S5:Click “Updates Names” button

```
$ ll ↓
```



Update standard folders to current language?

You have logged in in a new language. You can automatically update the names of some standard folders in your home folder to match this language. The update would change the following folders:

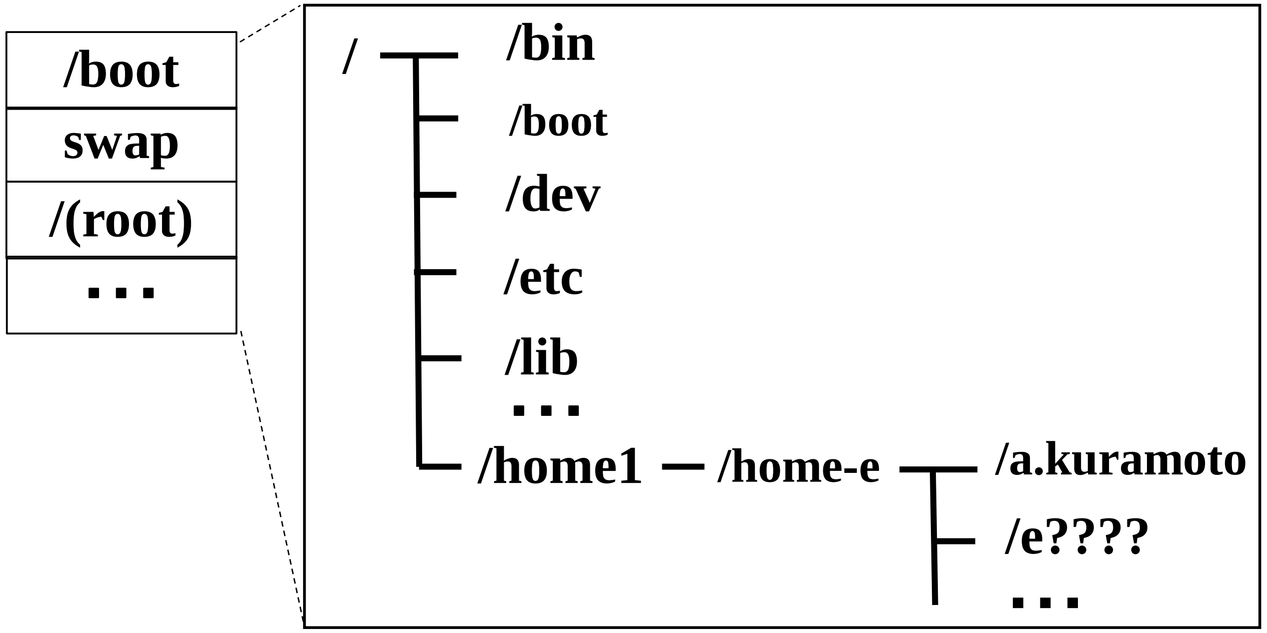
Current folder name	New folder name
/home1/home-e/a.kuramoto/デスクトップ	/home1/home-e/a.kuramoto/Desktop
/home1/home-e/a.kuramoto/ダウンロード	/home1/home-e/a.kuramoto/Downloads
/home1/home-e/a.kuramoto/テンプレート	/home1/home-e/a.kuramoto/Templates
/home1/home-e/a.kuramoto/公開	/home1/home-e/a.kuramoto/Public
/home1/home-e/a.kuramoto/ドキュメント	/home1/home-e/a.kuramoto/Documents
/home1/home-e/a.kuramoto/ミュージック	/home1/home-e/a.kuramoto/Music
/home1/home-e/a.kuramoto/ピクチャ	/home1/home-e/a.kuramoto/Pictures
/home1/home-e/a.kuramoto/ビデオ	/home1/home-e/a.kuramoto/Videos

Note that existing content will not be moved.

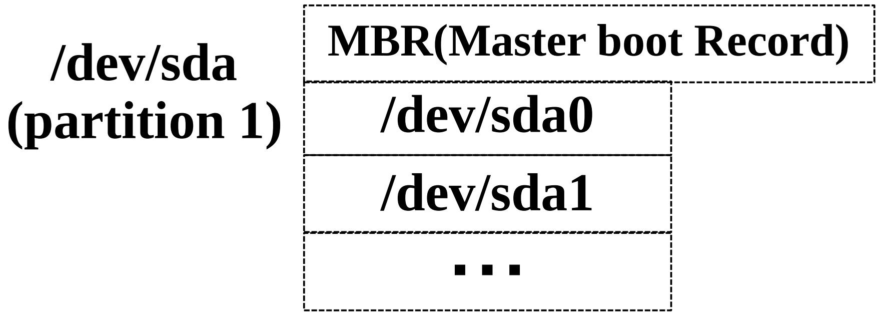
☐ Don't ask me this again

･Files and Directory Structure

【Disk partition】



【Disk Device】



Act1: check your home directory

$ *pwd* ↓	:Programmers working directory
$ *cd* ..↓	:Change directory to upper node
$ *ll* ↓	:List directory contents
$ cd / ↓	:Change directory to root
$ *ll* ↓	:See left diagram
$ cd ↓	:Move to home directory (HOMEDIR)

Act2: check your device and disk space

$ *d f* ↓	:Device free

･Linux File System(Types of files)

☆regular file

Anstructured byte strings stored in one dimension.

(ex) -rwxrwx--- 1 kuramoto ･･････clock0

☆directory

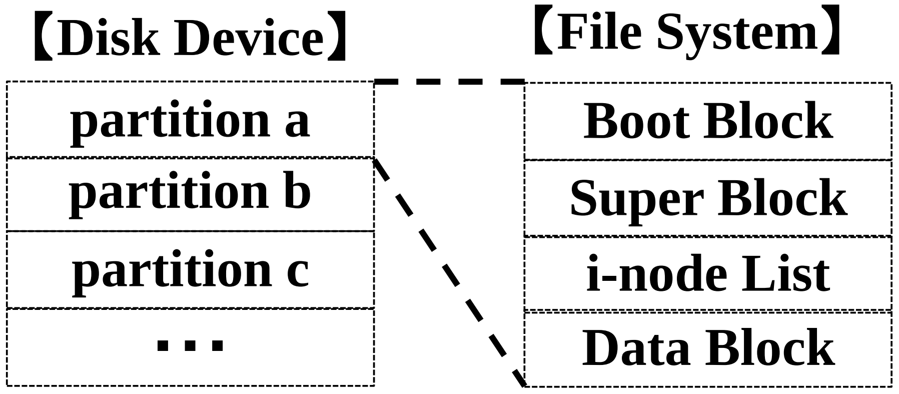
A file that associates a file name with a file body.

(ex) drwxrwxr-x 3 kuramoto ･････ prog2

☆special file

Unlike regular file,not used for storing data, but as a stream.

(ex) prw-rw-r-- 1 kuramoto ･･･ pipefile (FIFO file)
brw-rw---- 1 root disk ･･ sda (Block Device)
crw--w---- 1 root ･･ tty0 (Character Device)



Act3: Make a new directory

```
$ cd ↓            :Move to HOMEDIR
$ mkdir prog2     :Make "prog2" directory
$ cd prog2 ↓      :Move toprog2
$ ll ↓            :Check the directory
```

(Copy three files from Moodle System clock0.c , textDisp.c and gsh)

Act4: Check some spectial file attributes

```
$ mknod pipefile p ↓  :Make FIFO file
$ ll pipefile ↓
$ cd /dev ↓           :Move to dev directory
$ ll ↓                :Check the directory
$ cd ↓                :Move to HOMEDIR
$ ls -il ↓            :Check the i-node number
```

Optional Act: Move C source files from Desktop to "prog2" directory

```
$ cd ↓
$ cd デスクトップ↓
$ cp *.c prog2 ↓
$ ll ↓                :Check copied files
```

【Exercise1:clock0.c】Displaying current time on terminal using “time” system call (1)

```
// clock0. c
//
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include <string.h>
//
// Program start
//
int main(int argc, char **argv)
{
        time_t tt;                          //present time
        struct tm *ts;      //pointe to tm structure
        int h, m, s;        //Time . Minute . Second
        int yy, mm, dd;     //Year . Month . Day
        int ww;                             //Day of Week
        char yb[7][4] = {"Sun", "Mon", "Tue", "Wed", "Thu",
                            "Fri", "Sat"};
        char tstr [40];     // String for Time display
        char dstr [40];     // String for Date display

        while(1)            /// Forever Loop
        {
                time(&tt);                  // Current Time
                ts = localtime(&tt);        // time_t --> struct tm
                h = ts->tm_hour;            // Time
                m = ts->tm_min; // Minute
                s = ts->tm_sec;    // Second

                yy = ts->tm_year+1900;      // Year
                mm = ts->tm_mon+1;          // Month
                dd = ts->tm_mday;           // Day
                ww = ts->tm_wday;           // Day of Week
                strcpy(dstr, "");
                strcpy(tstr, "");
                // Day
                sprintf(dstr, "%04d/%02d/%02d (%s)", yy, mm, dd, yb[ww]);
                //Time
                sprintf(tstr, "%02d:%02d:%02d", h, m, s);
                printf("%s - %s¥n", dstr, tstr);
                sleep (1);
                }
                return(0);
}
```

```
$ cd  prog2 ↓
$ chmod 777 gsh ↓          :Change mode to executable
$ ./gsh clock0 ↓           :Compile “clock0.c” and make
                             executable file “clock0”
$ ./clock0 ↓                :Check copied files
2018/05/17 (Thu) - 13:30:48
2018/05/17 (Thu) - 13:30:49        Ctrl + c
```

【Exercise2:textDisp.c】(2)

```
// Edit textDisp.c
//
#include <GL/glut.h>
#include <stdio.h>
#include <string.h> // strlen()

void Display(void);
void Reshape(int,int);
void Timer(int);
void PrintText(int, int, char *);

char message[]="Welcome to Glut World!";
int counter=0;
//
int main(int argc, char **argv)
{
        glutInit(&argc, argv);
        glutInitWindowSize(500, 250);
        glutCreateWindow("Testing PrintText()") ;
        glutDisplayFunc(Display);
        glutReshapeFunc(Reshape);
        glutTimerFunc(1000,Timer,0);
        glutInitDisplayMode(GLUT_RGBA);
        glClearColor(0.0, 0.0, 0.0, 1.0);

        glutMainLoop();
```

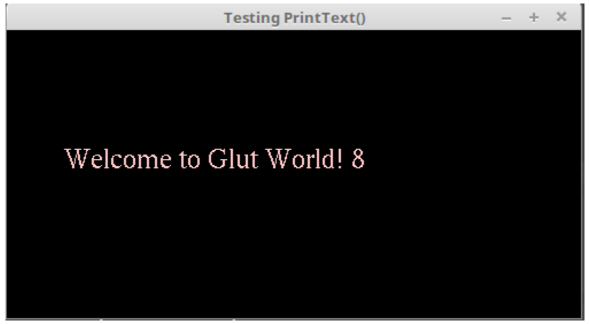
```
        return(0);
}
//
// Update Window Display
//
void Display(void)
{
        static    char num[5];

        glClear(GL_COLOR_BUFFER_BIT);

        glColor3ub(255, 200, 200);
        PrintText(50, 120, message);

        sprintf(num,"%d",counter);
//      glColor3ub(255, 200, 200);
        PrintText(300, 120, num);

        glFlush();
}
//
// Update Coordinate if Window Size Update occured
//
void Reshape(int w, int h)
{
```

【Exercise2:textDisp.c】(3)

```
		glViewport (0, 0, w, h);
		glMatrixMode(GL_MODELVIEW);
		glLoadIdentity();
		gluOrtho2D(0 , w, 0, h);
		glScaled (1, -1, 1);
		glTranslated (0, -h, 0);
		glutReshapeWindow (500 , 250);
					// Fixing Window Size
}
// Display Strings
//
void PrintText(int x, int y, char *s)
{
		int i , l;

		glRasterPos2i(x , y);

		l = strlen(s);

		for (i = 0 ; i < l ; i++)
		{
		glutBitmapCharacter(GLUT_BITMAP_TIMES
					_ROMAN_24, s[ i ]);
		}
}
//
// Timer Interrupt
```

```
void Timer(int value)
{
		counter++;
		glutPostRedisplay();
		glutTimerFunc(1000, Timer, value);
}
```

```
$ ./gsh textDisp ↓		:Compile "textDisp.c" and make
				 executable file "textDisp"
$ ./textDisp &↓			:Execute on background
```


Testing PrintText()

Welcome to Glut World! 8

```
$ ps ↓					:List Process IDs
PID TTY        TIME CMD
3045 pts/1   00:00:00 bash
4588 pts/1   00:01:05 textDisp  :4588 is the PID of textDisp
4600 pts/1   00:00:00 ps
$ kill  -9  4588↓  :terminate textDisp forcibly
```

【Exercise3:rectDraw.c】Displaying rectangle(square)

```
#include <GL/glut.h>

GLdouble s = 0.5;     ①

void display(void) {   ②
   /* Fil the window defined color*/
   glClear(GL_COLOR_BUFFER_BIT);   ③

   glColor3d(1.0, 0.0, 0.0);   ④

   glBegin(GL_LINE_LOOP);   ⑤

   glVertex2d(-s,-s);      ⑥
   glVertex2d(-s, s);
   glVertex2d( s, s);
   glVertex2d( s,-s);

   glEnd();

    glFlush();       ⑦
}
```

① Define Size of square

② Execute draw function

③ Fil the window defined color

④ Define red line

⑤ Specify primitive draw type as GL_LINE_LOOP

⑥ Define vertex coordinates

⑦ Flush all commands remained

【Exercise3:rectDraw.c】Displaying rectangle(square)

```
int main(int argc, char **argv){
    glutInit(&argc, argv);   ⑧

    glutInitWindowPosition(100, 100);   ⑨
    glutInitWindowSize(400, 400);   ⑩

    glutCreateWindow(“test”);   ⑪

    glutInitDisplayMode(GLUT_RGBA);   ⑫
    glClearColor(1.0, 1.0, 1.0, 1.0);

    glutDisplayFunc(display);  ⑬

    glutMainLoop();
    return 0;
}
```

⑧Initialaize GLUT, penGL.   Pass the arguments of main function as it is.

⑨ Define the position of window

⑩ Define the size of window

⑪Create window named “Sample 2”

⑫ Define background color

⑬ Specify function when redisplay

⑭ GLUT display loop infinitely

```
$ ./gsh rectDraw ↓
$ ./rectDraw & ↓
```

【Appendix 1】(Argument of glBegin)

Type	meaning
GL_Points	Displaydots
GL_LINES	Draw al line from start to end
GL_LINE_STRIP	Tie points in order
GL_LINE_LOOP	Tie points inorder and joint start and end
GL_TRIANGLE	Draw triangle of three vertexes
GL_QUADS	Draw rectangle of four vertexes
GL_TRIANGLE_STRIP	Continuously draw a triangle while sharing one side
GL_QUAD_STRIP	Continuously draw a rectangle while sharing one side
GL_TRIANGLE_FAN	Draw a triangle in a fan shape while sharing one side
GL_POLYGON	Draw a polygon that is the designated vertex

【Appendix 2】(Texteditor)

```
$ nano ↓
```

※ *If you familier to Lunux OS , vi editor is moreconvinient.*

【Problem】(System requirements)

・Display current time on GLUT window

・Use function “glutTimerFunc()” making cyclic interrupts