

A Complete Course in Computers - First Edition Primer

Manas Kalia and Rajat Kalia

7 December, 2017



Part I

Todarmal (A search Engine)

The project was later renamed tobu (with a new noticeboard image) in it's later run at IIT Delhi while i was the Administrator. Near closure , it was handled by “kala” guy for one day owing to his bullying while on that day also it was installed by me only although he requested the technique of setting up the project be explained to him. Our team totally discredits them as all the contact email-id's had other names that day(As per law code can't be copyrighted). In my presence , the ownership circulated amongst Shitiz Bansal, Vikas Rana ,Rahul Jain(of Jwala), 2 Vivek Maliks (one Raj(original amongst those two)and one Pig "Nishant Kumar"(the copy)) and Vivek Shrivastav although Vivek was the name they were taking all the time, probably of Niraj Patel as setting the project was difficult and it was not possible for those people without Niraj Patel to set it up on the server (if crashed).

Programmed by Bill (Babbar Bhangoo) and disscsed by him at Nahan, Problem Question worked on by Rajat (kalia). (2002 onwards)

```
//The driver.c file of Todarmal project to be stored in "src" folder

#include "myutil.h"
#include <sys/stat.h>
#include <dirent.h>

void rem_spaces(char *str) {
    int flag,i,j;
    // remove initial spaces
    for (flag=i=j=0;i<strlen(str);i++)
    {
        if (isspace(str[i]) && flag==0)
            continue;

        flag=1;
        str[j++]=str[i];
    }
    str[j]='\0';
    //remove trailing spaces
    for (i=strlen(str)-1;i>=0;i--)
    {
        if (isspace(str[i]))
            str[i]='\0';
        else break;
    }
}

int buildtree(char *share,char *name,char *ip, char *pid)
{
    char treefile[256],command[256],mpoint[256];
    int ret,success,pprotect;
    FILE *fp;

    success=pprotect=0;
    // pprotect = password protect
    strcpy(mpoint,"/mnt/tmp/mpoint_");
    strcat(mpoint,pid);
    strcpy(treefile,"../files/tmp/tree_");
    strcat(treefile,pid);
    ret=mkdir(mpoint,S_IRUSR | S_IWUSR | S_IXUSR);
    if (ret<0)
        perror("Error in creating mount pt ");
    strcpy(command,"/bin/tobumount ");
    strcat(command,ip);
    strcat(command," \"");
    strcat(command,name);
    strcat(command," \"");
    strcat(command,share);
    strcat(command," \"");
    strcat(command,pid);
    fp=popen(command,"r");
    // mount and see output of smbmount
    if (fp==NULL)
        perror("smbmount pipe error ");
    while (fgets(command,256,fp)!=NULL)
    {
        // if ((char *) strstr(command,"ERRbadpw")!=NULL
        // || (char *) strstr(command,"ERRnoaccess")!=NULL)
        // if ((char *) strstr(command,"ERRbadpw")!=NULL)
        pprotect=1;
        // has got some passwd
        else printf("SMBMNTOUT : %s",command);
    }
    // if (pprotect==1)
    // printf("%s (%s) -> %s is passwd protected\n",ip,name,share);
    pclose(fp);
    fp=fopen("/proc/mounts","r");
    // read /proc/mounts to see if u have been able to mount the share
    if (fp==NULL)
        perror("Error Can't read /proc/mounts ");
    while (fgets(command,256,fp)!=NULL)
    {
        if ((char *) strstr(command,pid)!=NULL)
            success=1; // so we have mounted that share
    }
    fclose(fp);
    if (success || pprotect)
    {
        if (success==1)
        {
            printf("Building tree %s (%s) -> %s\n",ip,name,share);
            fflush(stdout);
        }
        strcpy(command,"./tree ");
        strcat(command,mpoint); // local mount point
        strcat(command," > ");
        strcat(command,treefile);
        system(command); // build tree
        strcpy(command,"/bin/tobufree "); // a root setuid script
        strcat(command,pid);
        if (success==1)
        {
            system(command); // umount mpoint
        }
    }

    ret=rmdir(mpoint); // remove local mount pt
    if (ret<0)
        perror("Error in removing mount pt ");
}
```

```

        return(success || pprotect);
    }

    void writename(char *name,char *ip)
    {
        /* takes name and ip and writes name to file "../files/names/ip" */
        FILE *fp;
        char filename[256];
        strcpy(filename,"../files/names/");
        strcat(filename,ip);
        fp=fopen(filename,"w");
        if (fp==NULL)
        {
            fprintf(stderr,"Error Can't write name %s for %s",name,ip);
            perror("");
            return;
        }
        fprintf(fp,"%s",name);
        fclose(fp); }
    int findname(char *name,char *ip)
    {
        /*          * uses system command nmblookup to find name      * nmblookup is provided by smbclient
        * for details go to http://www.samba.org
        */
        FILE *pipe;
        char command[256],input[256],temp[5];
        int success=0,i;
        strcpy(command,"nmblookup -A ");
        // system command : nmblookup -A ip_address
        strcat(command,ip);      pipe=popen(command,"r");
        // run command and read output
        while(fgets(input,256,pipe)!=NULL)
        {
            if(((char *) strchr(input,'<')!=NULL)                                && ((char*) strchr(input,'>'))
                success=1; // so comp was connected
            if((char *) strstr(input,"<20>")!=NULL)
                continue; // skip lines not having <20>
                // process only lines with <20> in them
                // this line should have a name
            for(i=0;i<strlen(input);i++)
            {
                temp[0]=input[i];
                temp[1]=input[i+1];
                temp[2]=input[i+2];
                temp[3]=input[i+3];
                temp[4]='\0';
                if(strcmp(temp,"<20>")==0)
                {
                    input[i]='\0';
                    break;
                }
            } // input has still till <20> (excluding <20>)
            rem_spaces(input);
            strcpy(name,input); //we have name of comp now
        }
        pclose(pipe);
        if(success==1)
            writename(name,ip);
        return(success); }

    void writeshares(struct listnode *sharelist,char *ip) { /*          * if comp is connected ...      * then shares are writtern in ../files/shares/ip
    * and tree of shares is built in next stage      * this stage also deletes the tree files which correspond to ...      * shared folders which are no longer
    */
        FILE *fp;      DIR *dp;      struct dirent *dinfo;      char filename[256],*fileip,*fileshare,command[256];      struct listnode *p,*filelist;
        filelist=p=NULL;
        strcpy(filename,"../files/shares/");      strcat(filename,ip);      fp=fopen(filename,"w");      // write list of shares ...      if(fp==NULL)      {
    struct listnode *findshares(char *name,char *ip) {      struct listnode *sharelist=NULL;      char command[256],input[256];      FILE *pipe;      int success;
        strcpy(command,"smbclient -L \");      strcat(command,name);      strcat(command," -I ");      strcat(command,ip);      strcat(command," -N");
        pipe=popen(command,"r"); // command used as junk space below      while(fgets(input,256,pipe)!=NULL)      {      // add some smart error par
        // next time i will find the new name      // not doing it this time to avoidm race condition      success=1;      }
        || input[18]!='s' || input[19]!='k')      {      fprintf(stderr,"Error : \"Disk\" in %s\n",input);      continue;
    int main(int argc, char *argv[]) {      FILE *fp;      char ip[20],filename[256],name[20],sharename[256];      char pidjunk[20],command[256];      struct stat
        if(argc!=3)      {      fprintf(stderr,"Usage : driver ip_address sleep_time(min)\n");      exit(1);      }      ret=0;      ret=(int) stat
        if(strcmp(ip,"10.136.1.211")==0) exit(1);      time(&ctime); // current time
        strcpy(filename,"../files/names/");      strcat(filename,ip);      ret=stat(filename,&filebuf);      strcpy(name,"");      if((ret>=0) && (ctime - file
    && (strcmp(argv[2],"-1")!=0)){      // file for name is there and is not so old      fp=fopen(filename,"r");      if(fp==NULL)
        if(strcmp(name,"")==0)      {      writeshares(NULL,ip);      exit(1); // comp has no name      }
        // now that we have got name of comp      // let's find the shares of that comp      strcpy(filename,"../files/shares/");      strcat(filename,ip);      ret=
    && (strcmp(argv[2],"-1")!=0))      {      // we have info about shares      fp=fopen(filename,"r");      if(fp==NULL)      {
        // cool ... we have the share list .. time to build tree.      for(p=sharelist;p!=NULL;p=p->next)      {      strcpy(filename,"../files/t
    && (strcmp(argv[2],"-1")!=0))      continue; // skip this share      printf("Attempt to mount %s (%s) -> %s\n",ip,name,p->key);
        strcpy(command,"mv ");      strcat(command,filename); // tree file      strcat(command," \");      strcpy(filename,"../files/t
    freelist(sharelist); }

```


Part II

Puyo Puyo (A 2D Game)

This was the problem asked in my Gameloft Interview in the preliminary round. It was worked on by Rahul Agrawal. I chose PHP later in life and wasn't comfortable with Java. (2007)

```
import java.applet.*; import java.awt.*; import java.awt.image.*; import java.awt.event.*; //import java.net.*;
public class Puyopuyo extends Applet implements Runnable, KeyListener{ //drawing vars BufferedImage bufferdImg; Graphics2D bufferdImgSurface; Thread gameTh
//board vars public int bordXct=6,bordYct=12; //board dimensions //use only even numbers for bordXct... public int bordct=bordXct*b
public int bordY[] = new int[bordct]; //bord y public int bordpxlX[] = new int[bordct]; //xpxel position public int bordpxlY[] = new
public boolean bordcapsule[] = new boolean[bordct]; //is capsule public boolean bordfxd[] = new boolean[bordct]; //is fixed to bord? public in
public int iconz=32; //icon size public boolean canCtrlCaps; //can you control the capsule? public int capsDxn; //capsule d
/* capsDxn—— [pvt][odr] = 3
[odr][pvt] = 1
[pvt][odr] = 4
[odr][pvt] = 2
* the pvt being the Front of the capsule
1 = going left 2 = down 3 = ryt 4 = up
the number corresponds to these : and the number inside the plane corresponds to the index of array
_0_ _1_ _2_ _3_ _4_ _5_ 1 2 3 4 5 6 11 | 60 61 62 63 64
12 | 66 67 68 69 70 71 bord X
*/
//game timing vars public long dwnTymHoldr; //holder for the down tym public long dwnTym; //lenght of delay for icons to fall public l
//game switches boolean isOver; //game over boolean isAuto; //the game moves by itself boolean isStart; //start to play? boolean isF
//move swithces boolean toLft; //move capsule to left? boolean toRyt; //move capsule to ryt? boolean rot8L; //rot8 ccwise? boolean rot8R; //rot8 cwise
//score vars public long hiscore,score; public int lvl; public long lnkCtr;
//sleep vars i just copied this part from a tutorial but i know (barely) how it works // it's self explanator
//xtravars public int gg;
public void start(){ //i just copied this part from a tutorial :) Thread gameThread = new Thread(this); gameThread.start(); }
public void init(){ //i also copid this :) this is where initialization takes place int i; if (bordXct<6) // min size of vord
public void run(){ while(true){ // Starts the loop tick_start = System.currentTimeMillis(); if (ismovedelay && S
} repaint(); // Redraw the screen tick_end = System.currentTimeMillis(); tick_duration = tick_end - tick_start;
public void paint(Graphics g){ update(g); }
public void update(Graphics g){ //updates the screen if (freeze) //if is over, no mre... return; Graphics2D g2 = (Graphics2D
//drwCap(bufferdImgSurface); drwImg(bufferdImgSurface); //drwas the entire bord }
if (isPlay){ if (isAuto){ //the game automatically controls the game dwnTym=50; //makes down tym a bit faste
dwnTym=dwnTymHoldr; //dwnTym is reset to initialized value dwnTymStrt=System.currentTimeMillis(); //set dwnTym start
for (i=1;i<=Lnkct;i++){ //loop all links check every link if (chkbordlnk(i)){ // if there is something to
} else{ //break those links :) and make the others float for (i=1;i<=
} lnkCtr++; makeFloat(); //makes the PROBABLE icons float(un
toDwn=false; fin=true; } drwImg(bufferdImgSurface); else //true, u can contr
} if (fin){ //game over? //for (i=0;i<bordct;i++) //loop all icon // if (bordicon[i]!=0) //if not bla
public void initBoard(){ //init the bord int i; canCtrlCaps=false; capsDxn=0; bordPX=iconz*bordXct; bordPY=iconz*bordYct; scrW
public void initGame(){ initBoard(); score=0; lvl=1; lnkCtr=0; dwnTymHoldr=2500; dwnTym=dwnTymHoldr; ismovedelay=false;
public void chkstage(){ if (lnkCtr>=5){ lvl++; if (dwnTymHoldr>500) dwnTymHoldr=dwnTymHoldr-500;
public int getscore(int ini){ int i; int ctr=0; for (i=0;i<bordct;i++){ if (bordLnk[i]==ini) //if linknumber of icon==sought-aft
public void drwscores(Graphics2D sorpes){ sorpes.setColor(Color.white); if (score>hiscore && score>5000) hiscore=score; sorpes.draw
: " + hiscore,10,10); sorpes.drawString("Score : " + score,10,20); if (lvl!=0) sorpes.drawString("Lvl. : " + lvl,10,bordTop-7)
public void drwCap(Graphics2D sorpes){ sorpes.drawImage(eyecon[cap1],(scrWD/2)+iconz,bordTop-iconz-5,this); sorpes.drawImage(eyecon[cap2],(scrWD/2)+(ic
public void drwImg(Graphics2D sorpes){ //drwas the icons int i; int kulay; for(i=0;i<bordct;i++){ //loop all icons kulay=bordic
public void moveCLft(){ int c1,c2,cx1,cy1,cx2,cy2; c1=getIdxpvt(); //get the index if the PVT of capsule c2=getIdxodr(); //get the index of t
if (cx2==1 || cx1==1) //if any icon on the left border of bord ucant move enimore return; if (cx1>cx2 && !bordfxd[c2-1]){ //ca
/* the same concept of moving applies when going ryt */
public void moveCRyt(){ int c1,c2,cx1,cy1,cx2,cy2; c1=getIdxpvt(); c2=getIdxodr(); cx1=bordX[c1]; cy1=bordY[c1]; cx2=bordX[c
if (cx1==bordXct || cx2==bordXct) return; if (cx1>cx2 && !bordfxd[c1+1]){ //caps=horizontal .. the pvot is on the ryt && noth
public void switchCaps(){ //switch capsule color int c1,c2,cx1,cy1,cx2,cy2,tmp; c1=getIdxpvt(); //already discussed c2=getIdxodr(); cx1=
public void moverot8R(){ //rotate the capsule clockwise int c1,c2,cx1,cy1; c1=getIdxpvt(); //(:) c2=getIdxodr(); cx1=bordX[c1]; cy1=
if (capsDxn==1){ //ryt going down if (cy1!=bordYct){ //on topmost row? bordicon[c1+bordXct]=bordicon[c1]
/* same concept of rotating as moverot8R */ public void moverot8L(){ //rotate counter clockwise int c1,c2,cx1,cy1; c1=getIdxpvt(); c2=
if (capsDxn==3){ //left going down if (cy1!=bordYct){ bordicon[c1+bordXct]=bordicon[c2];
public void movecapDwn(){ //movecapsule down if (!canCtrlCaps) return; int c1,c2,cx1,cy1,cx2,cy2,tmp; boolean clu,c2u;
bordcapsule[c1]=false; } else{ //PVT under ODR if (chkUnder(cx1,cy1)) if (chkUnder(cx1,cy1))
public void clrLnk(){ //removes all link int i; Lnkct=0; for (i=0;i<bordct;i++) bordLnk[i]=0; //set link number to 0 }
public boolean chkbordlnk(int inLnknum){ //check if inLnknum is linked >4 .. inLnknum is a linknumber int i; int ctr=0; for (i=0;i<bordct;i
public void breaklnk(int inLnk){ //breaks the link of the linknumber used in conjunction with chkbordLnk int i; for (i=0;i<bordct;i++){
public void lnkBord(){ //link the bord clrLnk(); //clear links int x,y,idx,idxtgt; for (x=1;x<=bordXct;x++){ //loop to ry
Lnkct++; //create new link bordLnk[idx]=Lnkct; //and assign
bordLnk[idxtgt]=bordLnk[idx]; else if (bordLnk[idx]==0 && bordLnk[idxtgt]!=0) //idx have no link
bordLnk[idx]=bordLnk[idxtgt]; else{ //lnk not same make new link number
Lnkct++; //create new link bordLnk[idx]=Lnkct;
bordLnk[idxtgt]=bordLnk[idx]; else if (bordLnk[idx]==0 && bordLnk[idxtgt]!=0) //idx have no link
bordLnk[idx]=bordLnk[idxtgt]; else{ //lnk not same make new link number
// else //not fixd, floating or icon=0, no need to traverse, break to next x // break; }
}
public void lnkchange(int lnkfrm,int lnkto){ //change the link number of a into b .. self explanatory :) int i; for (i=0;i<bordct;i++){
public void moveallDwn(){ //move all icons down int i; for (i=0;i<bordct;i++){ if (!bordfxd[i] && bordicon[i]!=0){ //if
public void makeFloat(){ //chacks the highest pt of each columns and make the remaining float int x,y,z; boolean nomore; for (x=1;x<=
public void newCaps(){ //generates a new capsule cap1=getRandIcon(); cap2=getRandIcon(); isderCaps=true; }
public boolean transCaps(){ //transfercaps to bord capsDxn=3; int x,y,z; z=bordXct/2; x=getbordIdx(z,bordYct); if (bordfxd[x]) //a
return false;
bordpvt[x]=true; //make this PVT bordcapsule[x]=true; //and caps bordicon[x]=cap1; //get a random icon color bor
public int getIdxpvt(){ //gets the index of PVT int i; for (i=0;i<bordct;i++){ if (bordpvt[i] && bordcapsule[i]) //if this is
public int getIdxodr(){ //get ODR index int i; for (i=0;i<bordct;i++){ if (!bordpvt[i] && bordcapsule[i]) //this is not PVT "
public int getRandIcon(){ //(:) return (int)((Math.random()*100)%4)+1; }
public boolean chkFloat(){ //check if there are still floating int i; for (i=0;i<bordct;i++){ if (bordicon[i]!=0 && !bordfxd[i])
public int getbordIdx(int inX, int inY){ //returns the index of (x,y) return (((inY-1)*bordXct)+inX)-1; }
public boolean chkUnder(int inbordX,int inbordY){ //check if under of (x,y) is fixed or floor if (inbordY==1) //on lowest row? retu
public void moveiconDwn(int inX, int inY){ //moves the icon of current (x,y) down int lowbrd,bordidx; lowbrd=inY-1; bordidx=getbordIdx(inX,lowbr
public void drwBorder(Graphics2D sorpes){ //draws the lines //vertical drwLine(bordLft-3,bordTop-3,bordLft-3,bordTop+bordPY+3,sorpes);
public void drwLine(int X,int Y,int x,int y,Graphics2D sorpes){ //print borderlines if (x==X){ //vertical line sorpes.setColor(Color.white
sorpes.setColor(Color.gray); sorpes.drawLine(X+1,Y+1,x+1,y-1); sorpes.drawLine(X+2,Y+2,x+2,y-2); } if (y==Y){
public void keyPressed(KeyEvent ke){ int kcode=gg=ke.getKeyCode(); //get the key code if (!ismovedelay && canCtrlCaps && isPlay){ //if u can
```


Part III

Rootkit-lms v3 (learning management system of a study institute, Alpha
classes in this particular case)

(2009-2013)

Part IV

Vik the Legend (A 3D Game)

(2018 via Unity) The earth was made working by Taya Tayi (Jagjeet Manjeet)

Part V

A junior project (init function of a JQuery html design)

