LOAD DISTRIBUTION ON A LINUX CLUSTER

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

P-659

BACHELOR OF ENGINEERING

OF BHARATHIAR UNIVERSITY,

COIMBATORE.

Submitted by

S.ASHWIN KUMAR
JOY OUSEPH
A.PACKIARAJ
A.PRAKASH

P-659

Guided by

Mrs.N.SUGANTHI M.E

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641 006

MARCH 2002

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University, Coimbatore)

CERTIFICATE

This is to certify that project report entitled

LOAD DISTRIBUTION ON A LINUX CLUSTER

Is a bonafide record of work done by

S ASHWIN KUMAR	9827KO164
JOY OUSEPH	9827KO177
A PACKIARAJ	9827KO192
A PRAKASH	9827KO194

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

[illegible] 12/3/02
Head Of the Department

[illegible]
Staff-in-charge

Submitted for the University Examination held on__________

[illegible]
Internal Examiner

[illegible]
External Examiner

Place : Coimbatore
Date :

Dedicated to our beloved parents

and Friends

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We would like to begin with a special note of gratitude to our Principal **Dr.K.K.Padmanaban**, B.Sc. (Engg), M.Tech., Ph.D., and our Head of Department Prof. **Dr.S.Thangaswamy** Ph.D for their encouragement and making available to us the much needed facilities for successfully completing this project.

We also take immense pleasure in thanking our project coordinator **Mrs. S. Devaki** , M.S Assistant professor, without whose encouragement our endeavor wouldn't have been successful.

We are eternally indebted to our project guide Mrs. **N.Suganthi** M.E. who has been a constant source of inspiration and guidance. Her gentle yet firm goading helped us finish our project on time.

Our sincere gratitude to our parents, all the staff members and our dear friends who all played small but important parts in helping us complete this project.

Above all, we owe our gratitude to the Almighty, for showering abundant blessings on us.

SYNOPSIS

CONCLUSION

CONCLUSION

The system we have developed does make computationally intensive process more time efficient than when it is run on a stand-alone machine. Though this does a basic distributed processing namely, sorting, it is only a skeletal framework and a lot more has to be done to make it a truly distributed system with a commercial viability. In situations where great computation power is essential and powerful computers aren't feasible, this framework can be appropriately modified to suit individual application needs.

The undertaking of this project has provided us with an eye-opening experience to the possibilities in the world of cluster computing.

REFERENCES

REFERENCES

🕮 "Unix Networks programming" - Volume 1.

by W.Richard Stevens

Addison-Wesley publication

🕮 "Linux Application Development".

by Micheal and Johnson

Addison-Wesley publication

🕮 'Kdevelop' Application Development Tutorial.(KDE)

🕮 Qt Designer manual.(KDE)

APPENDIX

SAMPLE SOURCE CODE

```
//SERVER PROGRAM TO GIVE LOAD INFORMATION OF THE
//LOCAL MACHINE

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>
#include<sys/file.h>

#define MYPORT 3060
#include"defn.h"

    ser()
    {
      int sockfd, new_fd;
       pid_t pid;
      struct sockaddr_in my_addr,their_addr;
      int sin_size;
       float load;
       LOAD load_trans;
```

```
 if(fork()!=0)
   exit(0);
else
{
      if  ((sockfd  =  socket(AF_INET,  SOCK_STREAM,  0))  ==  -1)
      { perror("socket");
              exit(1);}



   my_addr.sin_family = AF_INET;
   my_addr.sin_port =  htons(MYPORT);
   my_addr.sin_addr.s_addr = INADDR_ANY;
   bzero(&(my_addr.sin_zero), 8);

    if    (bind(sockfd,    (struct    sockaddr    *)&my_addr,    sizeof(struct
    sockaddr)) == -1)
     {
        perror("bind");
        exit(1);
     }

    if (listen(sockfd, 5) == -1) {
    perror("listen");
    exit(1);
         }
    sin_size = sizeof(struct sockaddr_in);
      while(1)
        {

    if((     new_fd     =     accept(sockfd,NULL,NULL))==     -1)//(struct
sockaddr *)&their_addr,&sin_size)) == -1 )
```

```
            perror(" ACCEPT ERROR ");
            else
            {
           load_trans = load_find();
             if(send(new_fd,&load_trans.load,4,0) == -1)
                        printf("\n\n  Error in sending load");
            }
            close(new_fd);
      }
  } }
```

```
/********************** CLIENT PROGRAM*****************/

    #include <stdio.h>
    #include <stdlib.h>
    #include <unistd.h>
    #include <errno.h>
    #include <string.h>
    #include <netdb.h>
    #include <sys/types.h>
    #include <netinet/in.h>
    #include <sys/socket.h>
    #include <sys/file.h>
    #include"defn.h"

    #define PORT 3060

     main(int argc, char * argv[])
    {
       int sockfd, numbytes;
       char buf[20];
         float load;
         LOAD load_trans;
       FILE *fp;
       struct sockaddr_in their_addr; // connector's address information
       if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
         {
          perror("socket");
          exit(1);
       }
    their_addr.sin_family = AF_INET;
    their_addr.sin_port = htons(PORT);
    their_addr.sin_addr.s_addr = inet_addr(argv[1]) ;
    bzero(&(their_addr.sin_zero), 8);

      if( connect(sockfd,(struct sockaddr *)&their_addr,sizeof(struct sockaddr))
== -1)
    {
         perror("connect");
       exit(1);
    }
       else
       {
           load_trans.load = 5;

             /* Since there is a possibility of a number of
              * computers trying to write their load all at one
              * time ,file locking is used      */

             if(recv(sockfd,&load_trans.load,4,0) == -1)
                     perror("receive error ");
      fp = fopen("load_dat","a+");
             flock ( fileno ( fp ) , LOCK_EX ) ;
             fprintf(fp,"%s ",argv[1]);
             fprintf(fp,"%f\n",load_trans.load);
             flock ( fileno ( fp ) , LOCK_UN ) ;
             printf("IP Tested = %s\n",argv[1]);
      printf("\n  The  Retrived load = %f\n",load_trans.load);
        }
     close(sockfd);
   }
```

```
/***********************KDEVELOP FORM CODE-CPP FILE*********************/

#include "dlo.h"
#include"lclient.h"
#include<qmessagebox.h>
#include <qlabel.h>
#include <qpushbutton.h>
#include <qlayout.h>
#include <qvariant.h>
#include <qtooltip.h>
#include <qwhatsthis.h>
#include "dlf.h"


char *tm1,*tm2,buf1[40];
FILE *fp1;

dlo::dlo( QWidget* parent,  const char* name, bool modal, WFlags fl )
    : QDialog( parent, name, modal, fl )
{
    if ( !name )
        setName( "dlo" );
    resize( 510, 336 );
    QPalette pal;
    QColorGroup cg;
    cg.setColor( QColorGroup::Foreground, black );
    cg.setColor( QColorGroup::Button, QColor( 220, 220, 220) );
    cg.setColor( QColorGroup::Light, white );
    cg.setColor( QColorGroup::Midlight, QColor( 237, 237, 237) );
    cg.setColor( QColorGroup::Dark, QColor( 110, 110, 110) );
cg.setColor( QColorGroup::Mid, QColor( 146, 146, 146) );
cg.setColor( QColorGroup::Text, black );
cg.setColor( QColorGroup::BrightText, white );
cg.setColor( QColorGroup::ButtonText, black );
cg.setColor( QColorGroup::Base, white );
cg.setColor( QColorGroup::Background, QColor( 139, 139, 139) );
cg.setColor( QColorGroup::Shadow, black );
cg.setColor( QColorGroup::Highlight, QColor( 84, 112, 152) );
cg.setColor( QColorGroup::HighlightedText, white );
pal.setActive( cg );
cg.setColor( QColorGroup::Foreground, black );
cg.setColor( QColorGroup::Button, QColor( 220, 220, 220) );
cg.setColor( QColorGroup::Light, white );
cg.setColor( QColorGroup::Midlight, QColor( 253, 253, 253) );
cg.setColor( QColorGroup::Dark, QColor( 110, 110, 110) );
cg.setColor( QColorGroup::Mid, QColor( 146, 146, 146) );
cg.setColor( QColorGroup::Text, black );
cg.setColor( QColorGroup::BrightText, white );
cg.setColor( QColorGroup::ButtonText, black );
cg.setColor( QColorGroup::Base, white );
cg.setColor( QColorGroup::Background, QColor( 139, 139, 139) );
cg.setColor( QColorGroup::Shadow, black );
cg.setColor( QColorGroup::Highlight, QColor( 84, 112, 152) );
cg.setColor( QColorGroup::HighlightedText, white );
pal.setInactive( cg );
cg.setColor( QColorGroup::Foreground, QColor( 128, 128, 128) );
cg.setColor( QColorGroup::Button, QColor( 220, 220, 220) );
cg.setColor( QColorGroup::Light, white );
cg.setColor( QColorGroup::Midlight, QColor( 253, 253, 253) );
cg.setColor( QColorGroup::Dark, QColor( 110, 110, 110) );
cg.setColor( QColorGroup::Mid, QColor( 146, 146, 146) );
cg.setColor( QColorGroup::Text, black );
cg.setColor( QColorGroup::BrightText, white );
cg.setColor( QColorGroup::ButtonText, QColor( 128, 128, 128) );
cg.setColor( QColorGroup::Base, white );
cg.setColor( QColorGroup::Background, QColor( 139, 139, 139) );
cg.setColor( QColorGroup::Shadow, black );
cg.setColor( QColorGroup::Highlight, QColor( 84, 112, 152) );
cg.setColor( QColorGroup::HighlightedText, white );
pal.setDisabled( cg );
setPalette( pal );
setCaption( tr( "Load info" ) );
TextLabel9 = new QLabel( this, "TextLabel9" );
TextLabel9->setGeometry( QRect( 50, 90, 141, 31 ) );
QFont TextLabel9_font(  TextLabel9->font() );
TextLabel9_font.setFamily( "adobe-helvetica" );
TextLabel9_font.setPointSize( 12 );
TextLabel9_font.setBold( TRUE );
TextLabel9->setFont( TextLabel9_font );
TextLabel9->setText( tr( "IP ADDRESS" ) );
TextLabel10 = new QLabel( this, "TextLabel10" );
TextLabel10->setGeometry( QRect( 260, 90, 220, 30 ) );
QFont TextLabel10_font(  TextLabel10->font() );
TextLabel10_font.setFamily( "adobe-helvetica" );
TextLabel10_font.setPointSize( 12 );
TextLabel10_font.setBold( TRUE );
TextLabel10->setFont( TextLabel10_font );
TextLabel10->setText( tr( "PROCESSOR UTILIZATION (in %)" ) );

TextLabel13 = new QLabel( this, "TextLabel13" );
TextLabel13->setGeometry( QRect( 260, 150, 171, 30 ) );
QFont TextLabel13_font(  TextLabel13->font() );
```

```
TextLabel13_font.setFamily( "adobe-helvetica" );
TextLabel13_font.setPointSize( 12 );
TextLabel13_font.setBold( TRUE );
TextLabel13->setFont( TextLabel13_font );
TextLabel13->setText( tr( "TextLabel13" ) );
TextLabel14 = new QLabel( this, "TextLabel14" );
TextLabel14->setGeometry( QRect( 260, 220, 191, 30 ) );
QFont TextLabel14_font(  TextLabel14->font() );
TextLabel14_font.setFamily( "adobe-helvetica" );
TextLabel14_font.setPointSize( 12 );
TextLabel14_font.setBold( TRUE );
TextLabel14->setFont( TextLabel14_font );
TextLabel14->setText( tr( "TextLabel14" ) );
TextLabel11 = new QLabel( this, "TextLabel11" );
TextLabel11->setGeometry( QRect( 40, 150, 161, 31 ) );
QFont TextLabel11_font(  TextLabel11->font() );
TextLabel11_font.setFamily( "adobe-helvetica" );
TextLabel11_font.setPointSize( 12 );
TextLabel11_font.setBold( TRUE );
TextLabel11->setFont( TextLabel11_font );
TextLabel11->setText( tr( "TextLabel11" ) );
TextLabel12 = new QLabel( this, "TextLabel12" );
TextLabel12->setGeometry( QRect( 40, 210, 160, 41 ) );
QFont TextLabel12_font(  TextLabel12->font() );
TextLabel12_font.setFamily( "adobe-helvetica" );
TextLabel12_font.setPointSize( 12 );
TextLabel12_font.setBold( TRUE );
TextLabel12->setFont( TextLabel12_font );
TextLabel12->setText( tr( "TextLabel12" ) );

PushButton27 = new QPushButton( this, "PushButton27" );
```

```
    PushButton27->setGeometry( QRect( 390, 280, 100, 40 ) );
    QFont PushButton27_font(  PushButton27->font() );
    PushButton27_font.setFamily( "adobe-helvetica" );
    PushButton27_font.setPointSize( 12 );
    PushButton27_font.setBold( TRUE );
    PushButton27->setFont( PushButton27_font );
    PushButton27->setText( tr( "Get load" ) );

    TextLabel8 = new QLabel( this, "TextLabel8" );
    TextLabel8->setGeometry( QRect( 130, 20, 280, 41 ) );
    QFont TextLabel8_font(  TextLabel8->font() );
    TextLabel8_font.setFamily( "bookman" );
    TextLabel8_font.setPointSize( 16 );
    TextLabel8_font.setBold( TRUE );
    TextLabel8_font.setUnderline( TRUE );
    TextLabel8->setFont( TextLabel8_font );
    TextLabel8->setText( tr( "Load information across slaves" ) );

    connect( PushButton27, SIGNAL( clicked() ),this, SLOT( getload() ) );


}

dlo::~dlo()
{
}

bool dlo::event( QEvent* ev )
{
    bool ret = QDialog::event( ev );
```

```
if ( ev->type() == QEvent::ApplicationFontChange )
  {
     QFont TextLabel9_font(  TextLabel9->font() );
     TextLabel9_font.setFamily( "adobe-helvetica" );
     TextLabel9_font.setPointSize( 12 );
     TextLabel9_font.setBold( TRUE );
     TextLabel9->setFont( TextLabel9_font );
     QFont TextLabel10_font(  TextLabel10->font() );
     TextLabel10_font.setFamily( "adobe-helvetica" );
     TextLabel10_font.setPointSize( 12 );
     TextLabel10_font.setBold( TRUE );
     TextLabel10->setFont( TextLabel10_font );
     QFont TextLabel13_font(  TextLabel13->font() );
     TextLabel13_font.setFamily( "adobe-helvetica" );
     TextLabel13_font.setPointSize( 12 );
     TextLabel13_font.setBold( TRUE );
     TextLabel13->setFont( TextLabel13_font );
     QFont TextLabel14_font(  TextLabel14->font() );
     TextLabel14_font.setFamily( "adobe-helvetica" );
     TextLabel14_font.setPointSize( 12 );
     TextLabel14_font.setBold( TRUE );
     TextLabel14->setFont( TextLabel14_font );
     QFont TextLabel11_font(  TextLabel11->font() );
     TextLabel11_font.setFamily( "adobe-helvetica" );
     TextLabel11_font.setPointSize( 12 );
     TextLabel11_font.setBold( TRUE );
     TextLabel11->setFont( TextLabel11_font );
     QFont TextLabel12_font(  TextLabel12->font() );
     TextLabel12_font.setFamily( "adobe-helvetica" );
     TextLabel12_font.setPointSize( 12 );
     TextLabel12_font.setBold( TRUE );
```

```
        TextLabel12->setFont( TextLabel12_font );
        QFont PushButton27_font(  PushButton27->font() );
        PushButton27_font.setFamily( "adobe-helvetica" );
        PushButton27_font.setPointSize( 12 );
        PushButton27_font.setBold( TRUE );
        PushButton27->setFont( PushButton27_font );
        QFont TextLabel8_font(  TextLabel8->font() );
        TextLabel8_font.setFamily( "bookman" );
        TextLabel8_font.setPointSize( 16 );
        TextLabel8_font.setBold( TRUE );
        TextLabel8_font.setUnderline( TRUE );
        TextLabel8->setFont( TextLabel8_font );
    }
    return ret;
}

 void dlo::getload()
  {
   lclient();
   fp1=fopen("/proj/load_dat","r");

   fgets(buf1,40,fp1);
   tm1=strtok(buf1," ");
   tm2=strtok(NULL,"\n");

   TextLabel13->setText(tr(tm2));
   TextLabel11->setText(tr(tm1));

   fgets(buf1,40,fp1);
   tm1=strtok(buf1," ");
   tm2=strtok(NULL,"\n");
  TextLabel12->setText(tr(tm1));
  TextLabel14->setText(tr(tm2));

  fclose(fp1);

 QMessageBox::about(NULL,"Distribution","click to proceed");
  this->destroy;
  dlf *cli =new dlf();
  cli->show();


}

#include "dlo.moc"
//h FILE
#ifndef DLO_H
#define DLO_H

#include <qvariant.h>
#include <qdialog.h>
class QVBoxLayout;
class QHBoxLayout;
class QGridLayout;
class QLabel;
class QPushButton;
class dlo : public QDialog
{
   Q_OBJECT

public:
  dlo( QWidget* parent = 0, const char* name = 0, bool modal = FALSE, WFlags
  fl = 0 );
  ~dlo();

   QLabel* TextLabel9;
   QLabel* TextLabel10;
   QLabel* TextLabel13;
   QLabel* TextLabel14;
   QLabel* TextLabel11;
   QLabel* TextLabel12;
   QPushButton* PushButton27;
   QLabel* TextLabel8;

   public slots:
   virtual void getload();

 protected:
   bool event( QEvent* );
};

#endif // DLO_H




//CODE FOR SPLITTING


#include<stdio.h>
#include<math.h>
#include"round.h"
#include"intostr.h"
 int i=0,tempntm,nsystems,j=0;
```

```
         int lowest,iphigh,iplow,count=0,tlines=0,remaining=0;
         float load_total = 0,tempload_total = 0,exloadmachine;
         float temp2,sum;
        float hs,per,ls;
        char magni[10],command[100] ,*c;

struct load_info{
char ip[20];
float load;
float ntm;
int lines;
}load_arr[10];

FILE *fp,*fq,*fs;
char arr[40000][7] ,buf1[10],buf2[10];
int count = 0,take=0,i;
combine()
{
     fp=fopen("/proj/192.168.12.5.txt","r");
     fq=fopen("/proj/192.168.12.6.txt","r");
     fs=fopen("/proj/result.txt","w+");
     fscanf(fp,"%s",buf1);
    fscanf(fq,"%s",buf2);
     printf("%s   %s",buf1,buf2);
     if(strcmp(buf1,buf2)>0)

     {
         while(!feof(fq))
           {
              count++;
              fscanf(fq,"%s",buf2);
```

```
          fprintf(fs,"%s\n",buf2);
        }
        take=1;
}

else

{
        while(!feof(fp))
        {

           count++;
           fscanf(fp,"%s",buf1);
           fprintf(fs,"%s\n",buf1);
         }
        take=2;
 }

 if(take==1)
     {
          while(!feof(fp))
        {

            count++;
            fscanf(fp,"%s",buf1);
            fprintf(fs,"%s\n",buf1);
          }
       }

       else
        {
          while(!feof(fq))
         {

           count++;
           fscanf(fp,"%s",buf2);
           fprintf(fs,"%s\n",buf2);
           }

         }



 fclose(fp);
 fclose(fq);
 fclose(fs);

system("/proj/sort");
}



show()
{
            fp = fopen("load_dat","r");
           fq=fopen("first.txt","r");
           nsystems = 0;
         lowest =0 ;
         while(!feof(fp)){
         fscanf(fp,"%s",load_arr[i].ip);
         fscanf(fp,"%f",&load_arr[i].load);
         load_total += load_arr[i].load;
         if(load_arr[i].load < load_arr[lowest].load)
                lowest =i;
```

```
        printf("\n i = %d \t%s\t %f \n",i,load_arr[i].ip,load_arr[i].load);
        i++;
        nsystems++;
        }

printf("no of systems: %d\n",nsystems);

sum=load_arr[j].load+load_arr[j+1].load;
if(load_arr[j].load<load_arr[j+1].load)
  {
  hs=load_arr[j+1].load;
  ls= load_arr[j].load;
  iphigh=j+1;
  iplow=j;
   }
 else
   {
   hs=load_arr[j].load;
   ls=load_arr[j+1].load;
   iphigh=j+1;
   iplow=j;
   }
 per=(hs/sum)*100;

 printf("Total load=%f\n",sum);
 printf("highest=%f\n",hs);



 printf("lowest=%f\n\n",ls);
 printf("percentage1 =%f%\n",per);
 printf("percentage2 =%f%\n",(100-per));
```

```
while(!feof(fq))
  {
   fscanf(fq,"%s",magni);
   count++;
   }
printf("total no of lines: %d\n",(count-1));

tlines=(count-1)*(per/100);
remaining = count - (tlines+1);



printf("lines calculated for  %s   :%d\n",load_arr[iplow].ip,tlines);
printf("lines calculated for  %s :%d\n",load_arr[iphigh].ip,remaining);


}






 distribute()
  {
 strcpy(command,"");
 for(i=0;i<2;i++)
 {
     if (i==lowest)
     {
          if(strcmp(load_arr[i].ip,"192.168.12.5")!=0)
                    {
                         strcat(command,"rsh ");
                         strcat(command,load_arr[i].ip);
                         strcat(command," /proj/ex-1 ");
                    strcat(command," 0 ");
                    strcat(command,inttostr(tlines));


                         strcat(command," 192.168.12.6.txt");




                     printf("Client process :%s\n",command);
                }
          else
                {
                     strcat(command," /proj/ex-1 ");
                     strcat(command,inttostr(tlines+1));
                     strcat(command," ");
                     strcat(command,inttostr(count-1));
 strcat(command," 192.168.12.5.txt");

                     printf("Server proecss :%s\n",command);
                }


}


else
     {
                    if(strcmp(load_arr[i].ip,"192.168.12.5")!=0)
                {
                     strcat(command,"rsh ");
                     strcat(command,load_arr[i].ip);
                     strcat(command," /proj/ex-1 ");
                     strcat(command," 0 ");
                strcat(command,inttostr(tlines));
                strcat(command," 192.168.12.6.txt");
                printf("Client process :%s\n",command);
            }


             else
            {



                strcat(command," /proj/ex-1 ");
                strcat(command,inttostr(tlines+1));
                strcat(command," ");
                strcat(command,inttostr(count-1));
                strcat(command," 192.168.12.5.txt");
                printf("Server proecss :%s\n",command);
             }

           }



      if(!fork())
      {
        system(command);
        exit(0);
      }
      strcpy(command,"");
  }

printf("\nFinished.");
fclose(fq);
fclose(fp);
system(wc *192);
}
```

```
//CODE FOR GENERATING INPUT FILE

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

 main()
{
FILE *fs;
 int i ,j ,k,l,m,n,b,o,p;
   char temp[10];
    int count=0;

  char  a[10][2]= { "s","f","r","y","u","i","c","a","w","q"};
  fs=fopen("/proj/first.txt","w");

    strcpy(temp,"");

     for(i=0;i<10;i++)
     for(j=1;j<10;j++)
       for(k=2;k<10;k++)
         for(l=3;l<10;l++)
            for(m=4;m<10;m++)

             {

               strcat(temp,a[i]);
               strcat(temp,a[j]);
               strcat(temp,a[k]);
               strcat(temp,a[l]);
            strcat(temp,a[m]);
            printf("%s \n",temp);
            fprintf(fs,"%s \n",(const char *)temp);
            count++;
            strcpy(temp,"");
           }
        printf("COUNT = %d",count);
       fclose(fs);


          }
//CODE FOR SORTING
#include <stdlib.h>
#include <string.h>
#include "sort.h"

static int merge(void *data, int esize, int i, int j, int k, int (*compare)
  (const void *key1, const void *key2)) {

char             *a = data,
                 *m;

int             ipos,
                jpos,
                mpos;

ipos = i;
jpos = j + 1;
mpos = 0;


if ((m = (char *)malloc(esize * ((k - i) + 1))) == NULL)
```

```
  return -1;

while (ipos <= j || jpos <= k) {

  if (ipos > j) {

        while (jpos <= k) {

      memcpy(&m[mpos * esize], &a[jpos * esize], esize);
      jpos++;
      mpos++;

    }

    continue;

    }

  else if (jpos > k) {

        while (ipos <= j) {

      memcpy(&m[mpos * esize], &a[ipos * esize], esize);
      ipos++;
      mpos++;

    }

    continue;

  }
```

```
    if (compare(&a[ipos * esize], &a[jpos * esize]) < 0) {

    memcpy(&m[mpos * esize], &a[ipos * esize], esize);
    ipos++;
    mpos++;

    }

  else {

     memcpy(&m[mpos * esize], &a[jpos * esize], esize);
     jpos++;
     mpos++;

  }

}

memcpy(&a[i * esize], m, esize * ((k - i) + 1));

free(m);

return 0;

}

int mgsort(void *data, int size, int esize, int i, int k, int (*compare)
   (const void *key1, const void *key2)) {

int                j;
```

```
if (i < k) {

  j = (int)(((i + k - 1)) / 2);

    if (mgsort(data, size, esize, i, j, compare) < 0)
    return -1;

   if (mgsort(data, size, esize, j + 1, k, compare) < 0)
     return -1;



   if (merge(data, esize, i, j, k, compare) < 0)
     return -1;

 }

 return 0;

 }


 #include <stdio.h>
 #include <string.h>

 #include "sort.h"


 #define         STRSIZ              7
```

```
static void print_idata(const int *array, int size) {

int                i;

for (i = 0; i < size; i++)
   fprintf(stdout, "A[%02d]=%d\n", i, array[i]);

return;

}

static void print_sdata(char array[][STRSIZ], int size) {

int                i;



for (i = 0; i < size; i++)
   fprintf(stdout, "A[%02d]=%s\n", i, array[i]);

return;

}

static int compare_int(const void *int1, const void *int2) {



if (*(const int *)int1 > *(const int *)int2)
   return 1;
else if (*(const int *)int1 < *(const int *)int2)
   return -1;
else
```

```
  return 0;

}


static int compare_str(const void *str1, const void *str2) {

int             retval;

if ((retval = strcmp((const char *)str1, (const char *)str2)) > 0)
   return 1;
else if (retval < 0)
   return -1;
else
   return 0;

}


int main(int argc, char **argv) {


char              sarray[30240][STRSIZ] ,buf[7],path[25];

int              size ,start,end,i,j;

int count=0;

FILE *fp,*fq;
start=atoi(argv[1]);
end =atoi(argv[2]);
```

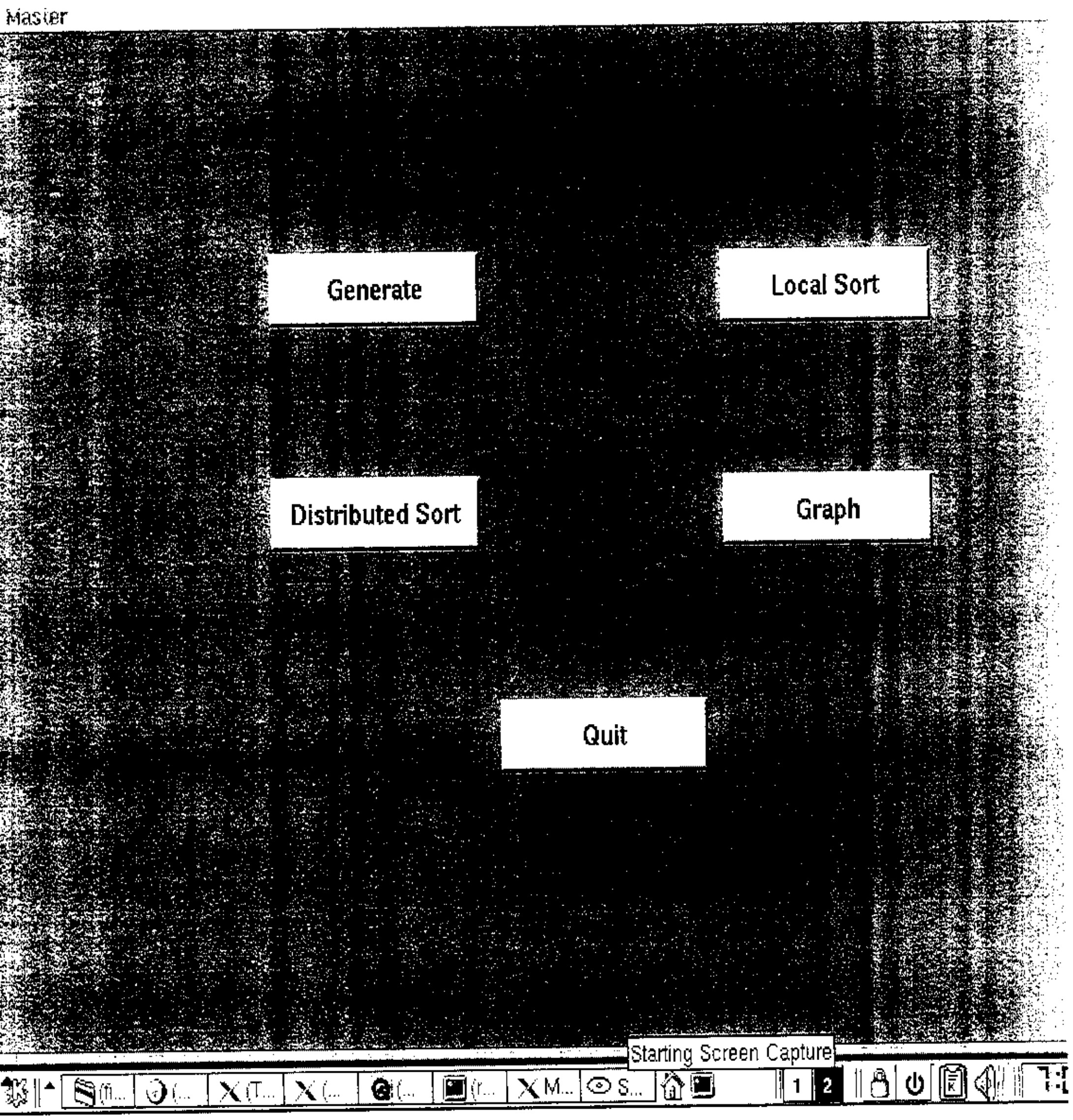
```
strcat(path,"/proj/");
strcat(path,argv[3]);
//printf("%s",path);
fp=fopen("/proj/first.txt","r");
fq=fopen(path,"w");
 size = end-start;
 //printf("%d\n",size);
 //printf("%d\n",count);
if(start==0)
 {
 while(count<size)
 {
  fscanf(fp,"%s",buf);
  //printf("%s",buf);
  strcpy(sarray[count++],buf);


 }
 }


 else
  {


    j=start*7;
    i=fseek(fp,j,SEEK_SET);
     while(count<size)
     {
    fscanf(fp,"%s",buf);
    strcpy(sarray[count++],buf);
    //printf("%s",buf);


   }
}



fprintf(stdout, "Before mgsort\n");
print_sdata(sarray, size);
printf("Sorting....\n");

if (mgsort(sarray, size, STRSIZ, 0, size - 1, compare_str) != 0)
   return 1;
fprintf(stdout, "After mgsort\n");
print_sdata(sarray, size);
printf("\nWriting...to %s",argv[3]);
for (i = 0; i < size; i++)
 fprintf(fq,"%s\n", sarray[i]);
 fclose(fp);
 fclose(fq);
return 0;

}
```

```
//CODING FOR COMBINING

#include<stdio.h>
#include<math.h>
#include"round.h"
#include"intostr.h"
#include<unistd.h>

struct load_info{
char ip[20];
float load;
float ntm;
int lines;
}load_arr[10];

FILE *ff,*fq,*fs;
char arr[40000][7] ,buff[10],buf2[10];
int take=0,c=0;
main()
{
     ff=fopen("/proj/192.168.12.5.txt","r");
     fq=fopen("/proj/192.168.12.6.txt","r");
     fs=fopen("/proj/result.txt","w+");
     fscanf(ff,"%s",buff);
    fscanf(fq,"%s",buf2);
     printf("%s   %s",buff,buf2);

    // getchar();
     if(strcmp(buff,buf2)>0)

     {
```

```
   while(!feof(fq))
     {
       c++;
       fscanf(fq,"%s",buf2);
       fprintf(fs,"%s\n",buf2);
      }
      take=1;
 }

 else

 {
      while(!feof(ff))
      {


        c++;
        fscanf(ff,"%s",buff);
        fprintf(fs,"%s\n",buff);
       }
      take=2;
  }

  if(take==1)
     {
        while(!feof(ff))
      {


         c++;
         fscanf(ff,"%s",buff);
         fprintf(fs,"%s\n",buff);
        }
        }


        else
         {
            while(!feof(fq))
          {


           c++;
           fscanf(ff,"%s",buf2);
           fprintf(fs,"%s\n",buf2);
           }


          }



 fclose(ff);
 fclose(fq);
 fclose(fs);

system("/proj/sort");
}
```

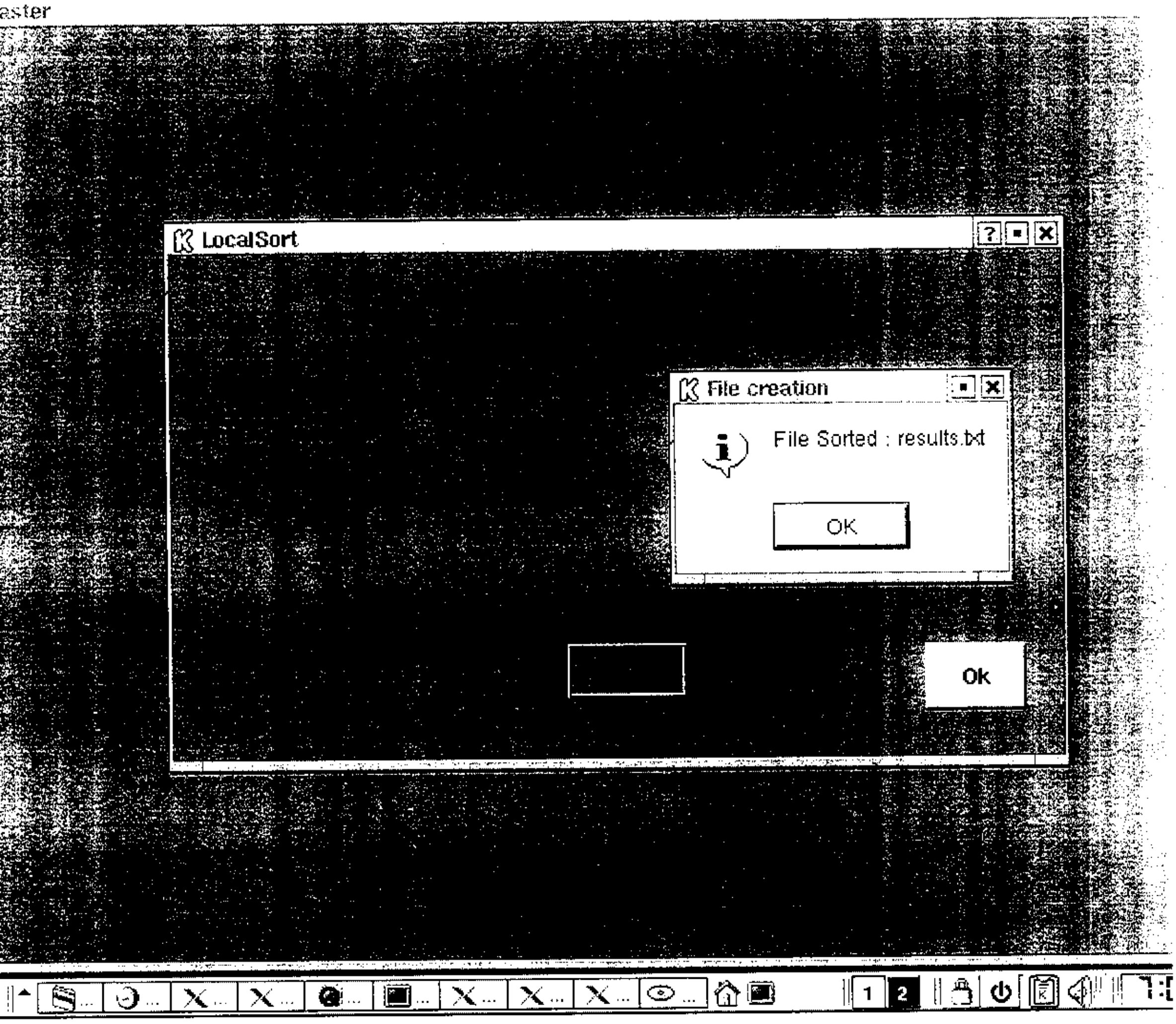
SAMPLE OUTPUT SCREEN

Master Screen



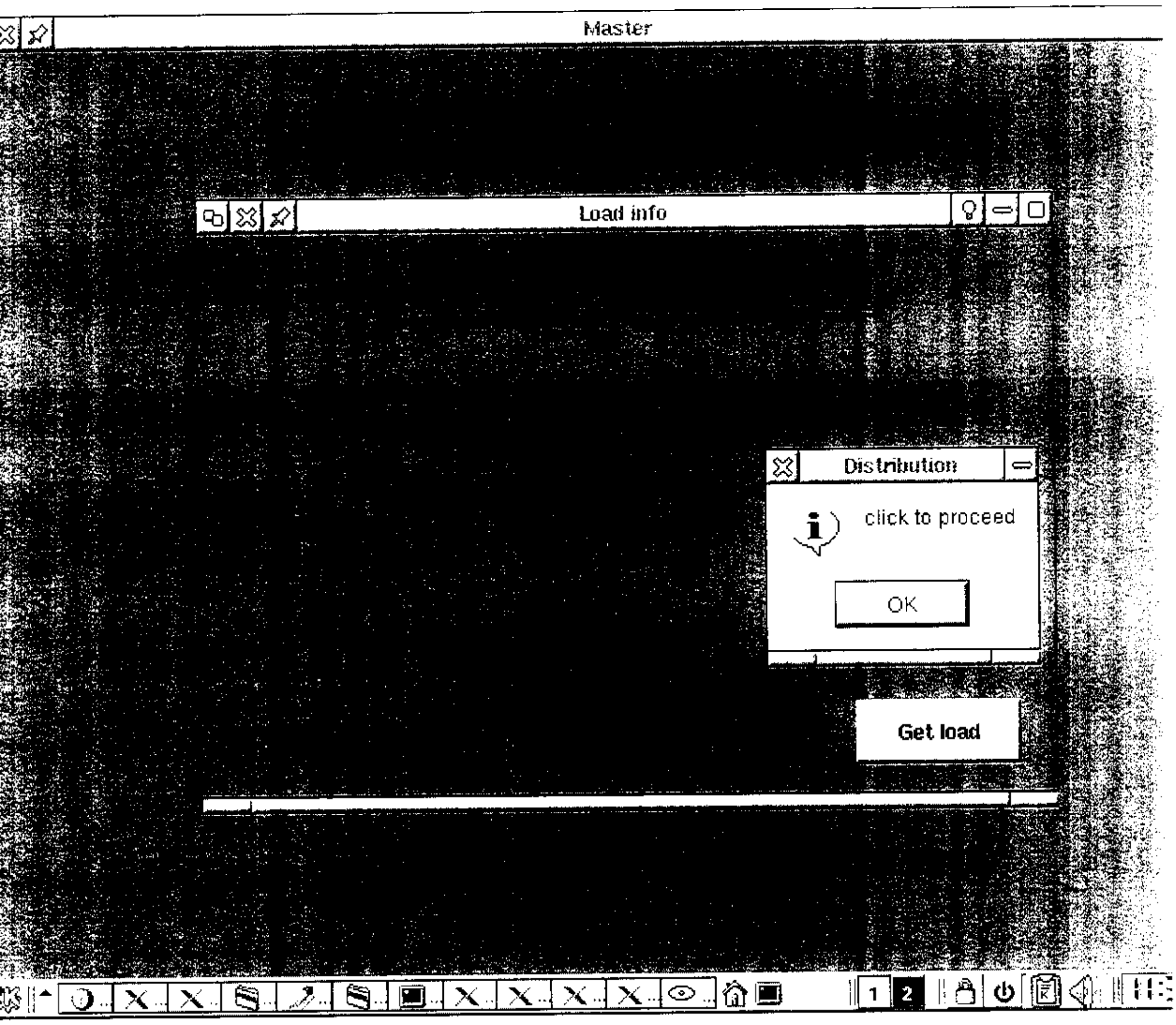
ocal Sorting

aster

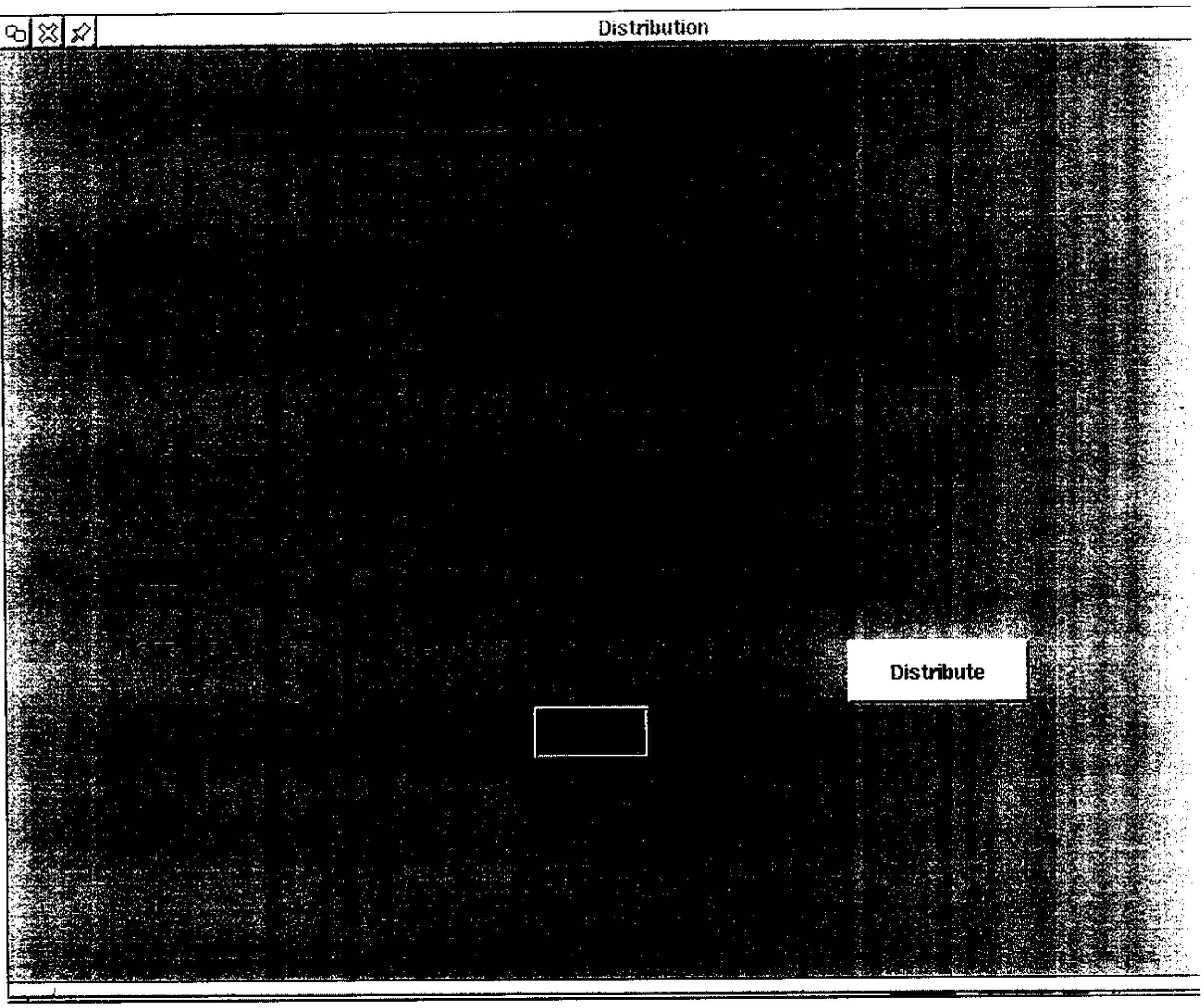


File Sorted : results.txt

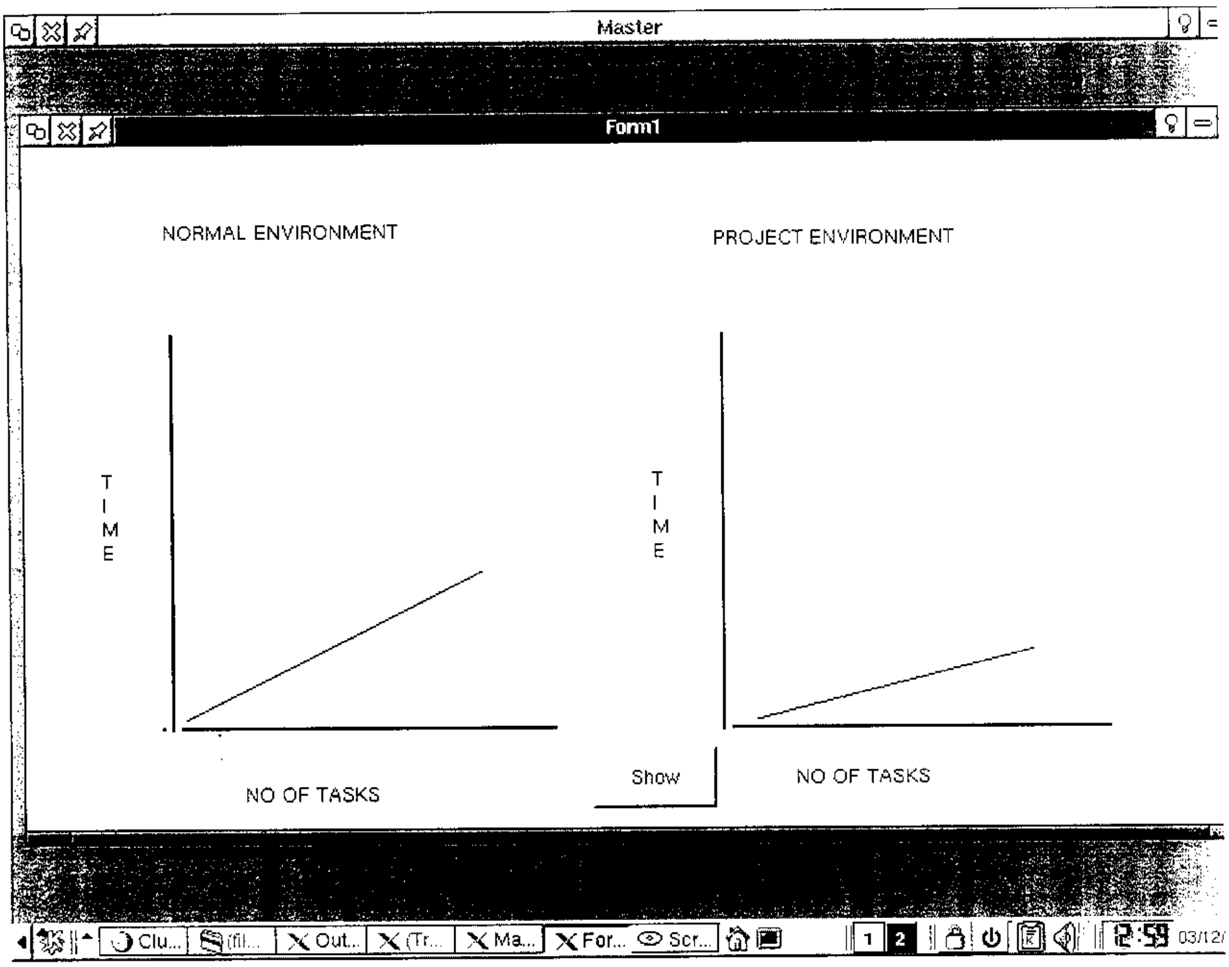
oad across Slaves



stributed sorting



Sample output


Master
Form1
NORMAL ENVIRONMENT
PROJECT ENVIRONMENT
TIME
TIME
NO OF TASKS
NO OF TASKS
Show
Clu...
(fil...
Out...
(Tr...
Ma...
For...
Scr...
1
2
12:59
03/12/