TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------

****

**Đề tài số 11**

BÁO CÁO BÀI TẬP LỚN

MÔN: Lập trình hệ thống

***Giảng viên hướng dẫn: Đặng Thành Phu***

Họ tên – Nhóm 11:Trần Bá Phúc– 1710A03

Nguyễn Thu Phương – 1810A04

Nguyễn Lý Quang – 1710A04

Hà Nội – 2020

MỤC LỤC

[1. Mở đầu 2](#_Toc40020321)

[2. Đề bài 3](#_Toc40020322)

[3. Mô tả bài tập lớn 4](#_Toc40020323)

[3.1 Chương trình C 4](#_Toc40020324)

[3.2 Chức năng GETTIME 6](#_Toc40020325)

[3.3 Chức năng vào một xâu và hiện ngược lại 7](#_Toc40020326)

[3.4 Chức năng kiểm tra số lượng kRAM trên mainboard 7](#_Toc40020327)

[4. Lập trình – cài đặt 7](#_Toc40020328)

[4.1 Chương trình chính 7](#_Toc40020329)

[4.2 GETTIME 10](#_Toc40020330)

[4.3 Vào một xâu và hiện ngược lại 11](#_Toc40020331)

[4.4 Kiểm tra số lượng kRAM trên mainboard 11](#_Toc40020332)

[5. Demo toàn bộ chương trình 12](#_Toc40020333)

[5.1 MENU chương trình 12](#_Toc40020334)

[5.2 GETTIME 13](#_Toc40020338)

[5.3 Vào một xâu và hiện ngược lại 14](#_Toc40020339)

[5.4 Kiểm tra số lượng kRAM trên mainboard 14](#_Toc40020340)

[6. Kết luận 15](#_Toc40020341)

[7. Tài liệu tham khảo 15](#_Toc40020342)

# MỞ ĐẦU

Lập trình hệ thống giúp cho sinh viên viết được chương trình bằng ngôn ngữ Assembly trên máy tính PC . Sinh viên có các kĩ năng đơn giản như : Sử dụng trình biên dịch hợp ngữ trong môi trường Window, sửa lỗi, liên kết, khảo sát tập lệnh, các ngắt đơn giản của hệ điều hành DOS. Để vận dụng và nâng cao được kĩ năng lập trình hệ thống bằng hợp ngữ, sinh viên phải nỗ lực rất nhiều trong việc tự học, bổ sung những kiến thức nhất định về phần cứng máy tính cũng như nguyên lý vận hành của các thiết bị ngoại vi có liên quan như : Máy in, hệ vi điều khiển, cổng vào ra nối tiếp/song song, …

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức, cũng như các công ty. Nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ. Việc lập trình lên các hệ thống để phục vụ cho các nhu cầu riêng của tổ chức, công ty thậm chí các cá nhân, không lấy gì làm xa lạ. Với một vài thao tác đơn giản, một người bất kì có thể sử dụng một hệ thống đơn giản.

# ĐỀ BÀI

Hãy viết chương trình thực hiện 4 chức năng :

Hãy viết chương trình thực hiện 4 chức năng :

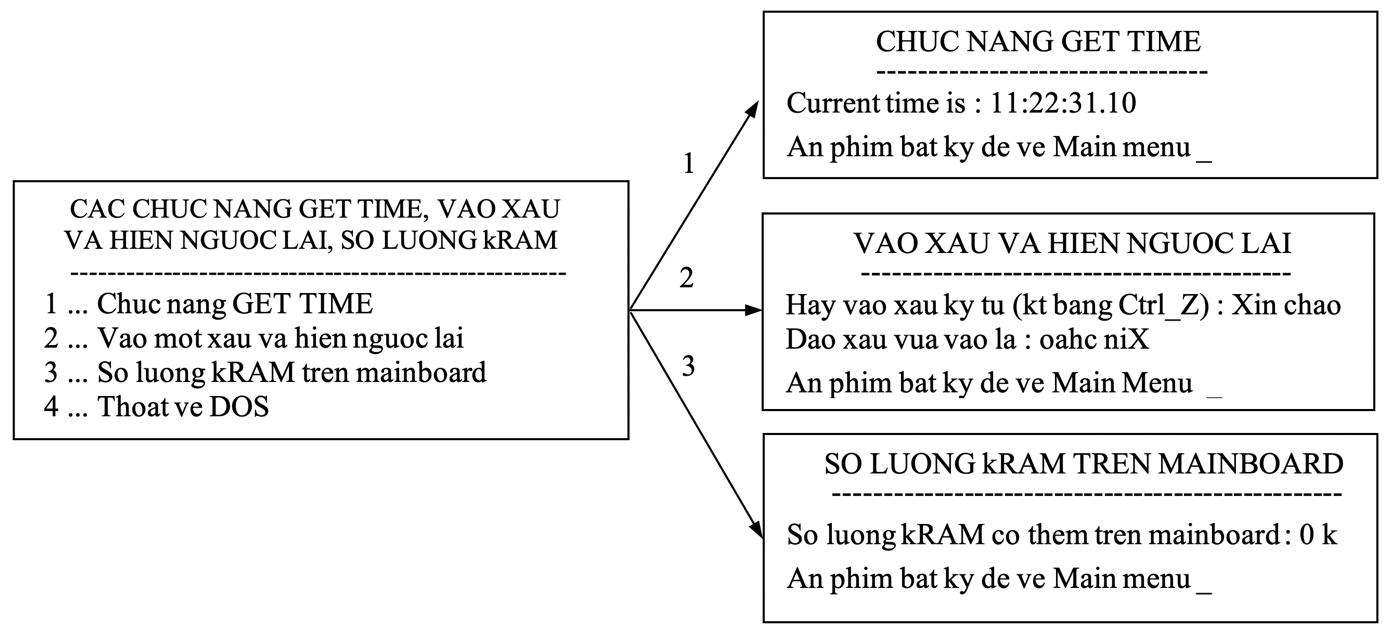
1 .... Lấy thời gian (time) của của hệ thống

2 .... Vào xâu ký tự, sau đó hiện ngược lại

3 .... Số lượng kRAM có thêm trên mainboard

4 ....Thoát khỏi chương trình

Ví dụ khi CT chạy :



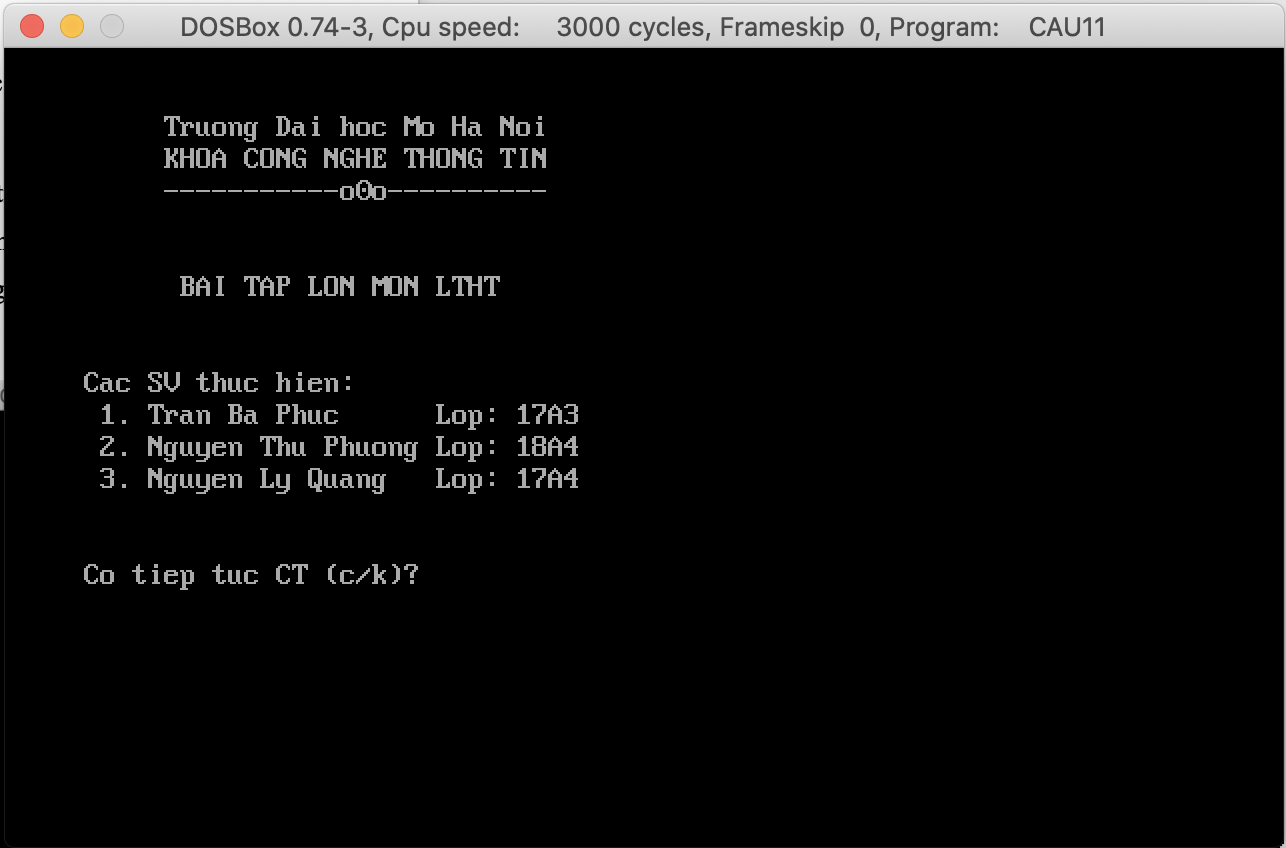
# MÔ TẢ BÀI TẬP LỚN

Đây là chương trình đa tệp, có sự liên kết giữa tệp C và tệp ASM. Trong đó, màn hình giới thiệu và màn hình chức năng do chương trình C viết, các chức năng cụ thể do chương trình con viết bằng ASM đảm nhiệm. Nhãn dùng chung là tên các chương trình con từ tệp ASM.

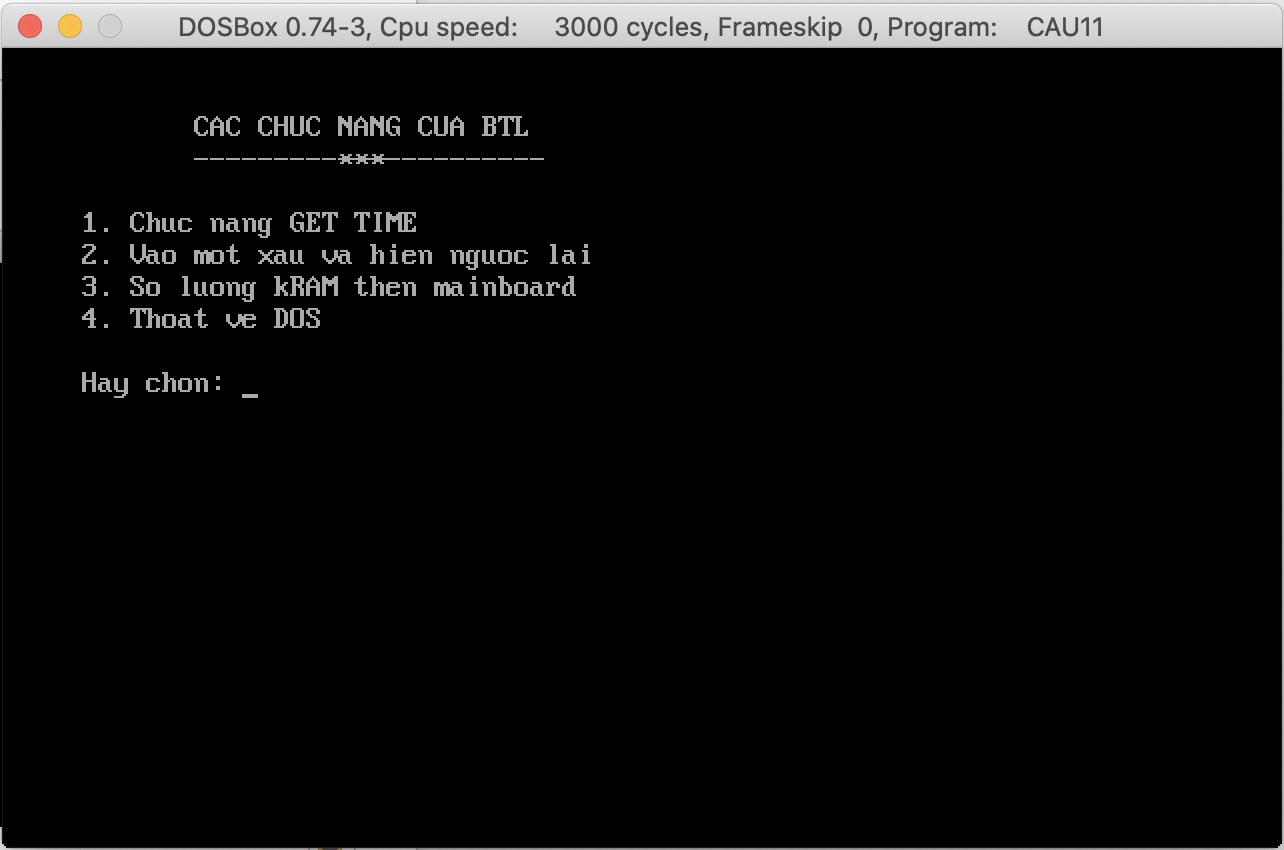
## Chương trình C

Cần khai báo extern các nhãn dùng chung là tên các chương trình con bao gồm: GETTIME, DAOCHUOI và KRAM.

Chương trình C sẽ hiện màn hình giới thiệu thông tin của sinh viên và lời nhắc ‘Có muốn tiếp tục chương trình không (c/k): ’. Nếu nhấn phím ‘c’ thì màn hình chức năng sẽ hiện ra, ngược lại thì thoát khỏi chương trình. Màn hình giới thiệu sẽ có nội dung như sau:



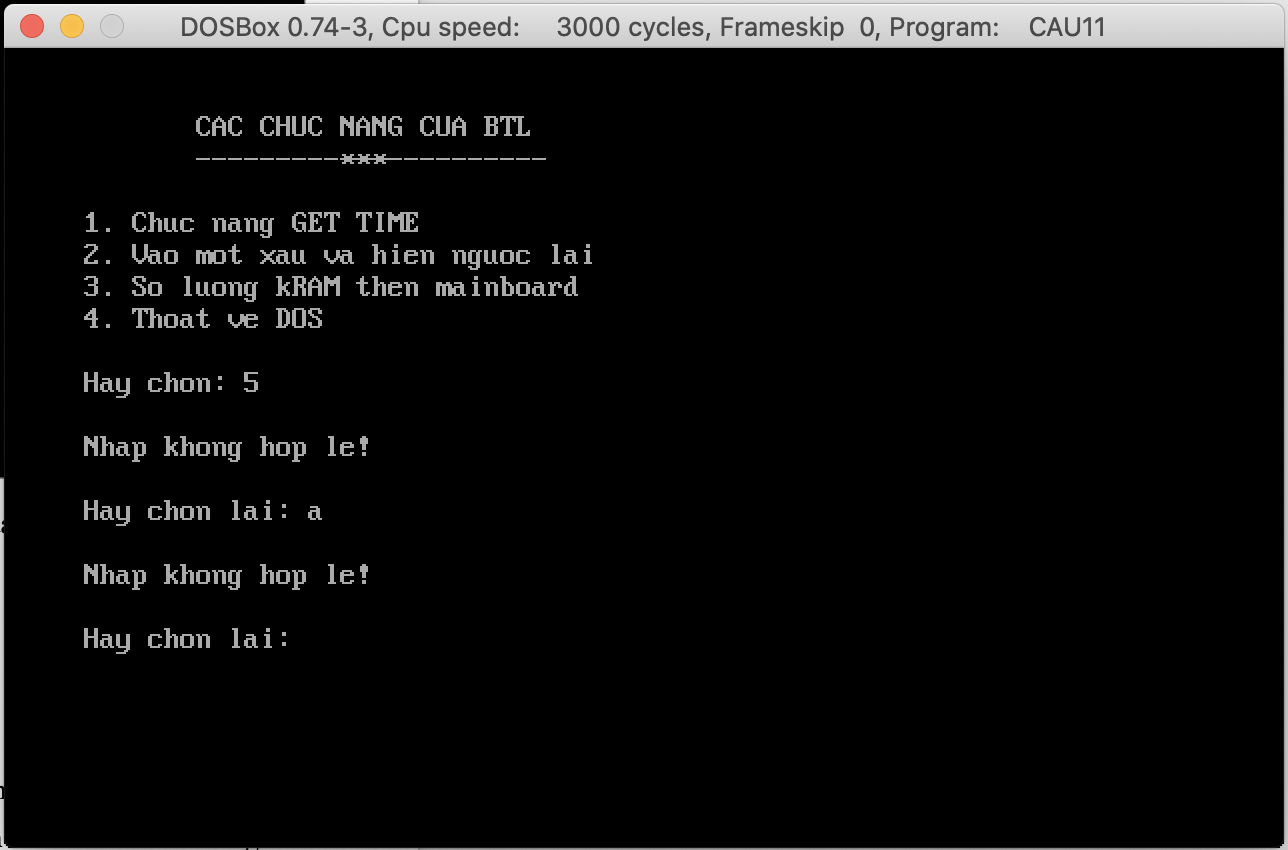
Nếu phím ‘c’ được nhấn, màn hình chức năng của chương trình sẽ có dạng:



Tại đây, sẽ thấy danh sách các chức năng chính của chương trình bao gồm:

* 1. Chức năng GETTIME
* 2. Vào một xâu và hiện ngược lại
* 3. Số lượng kRAM trên mainboard
* 4. Thoát về DOS

Muốn sử dụng chức năng nào thì chỉ cần nhấn phím số tương ứng của chức năng đó. Nếu bạn lỡ nhấn sai các giá trị của các phím từ 1 đến 4 sẽ có thông báo ‘Nhập không hợp lệ! ’, ‘Hãy chọn lại:’



Nhấn phím 4 để thoát khỏi chương trình. Ngược lại nếu chọn các phím chức năng từ 1 đến 3, chương trình C sẽ gọi chương trình con tương ứng do tệp ASM viết. Khi sử dụng xong sẽ thoát chương trình con và trở về MENU lựa chọn chính.

## Chức năng GETTIME

* Chức năng: Lấy thời gian (giờ, phút, giây và % giây) của máy tính
* Cách làm:
* Đầu tiên ta hiện thông báo “Giờ phút giây của máy tính”
* Để lấy time ta sử dụng chức năng **2ch** của ngắt int **21h**
* Sau khi dùng chức năng **2ch** của ngắt int **21h** thì giờ sẽ được chứa trong thanh ghi **ch**, phút sẽ chứa trong thanh ghi **cl**, giây chứa trong **dh** và phần trăm giây được chứa trong **dl**. Sau đó ta lần lượt hiển thị giá trị của các thanh ghi sau đó dùng Macro CALL\_HIEN\_SO\_N để hiển thị giờ trong **ax**, dùng Macro HienString để hiển thị thông báo hoặc các kí tự ngăn cách thành phần
* Các MACRO và chương trình con sử dụng:
  + HienString: hiện một xâu ký tự kết thúc bằng ‘$’ ra màn hình
  + clrscr: Xóa màn hình
  + HIEN\_SO\_N: hiện giá trị có trong **AX** ra màn hình

## Chức năng vào một xâu và hiện ngược lại

* Chức năng: nhập vào 1 xâu và hiện ngược lại
* Cách làm: Dùng cơ chế push và pop
* Ký tự nào push trước thì ra sau, pop sau ra trước
* Mỗi lần push thì ta đếm để dùng làm chỉ số trong CX
* Dùng loop để lặp đi lặp lại việc lấy và hiện ký tự
* Các MACRO và chương trình con sử dụng:
  + HienString: hiện một xâu ký tự kết thúc bằng ‘$’ ra màn hình
  + clrscr: Xóa màn hình

## Chức năng kiểm tra số lượng kRAM trên mainboard

* Chức năng: kiểm tra số lượng kRAM trên mainboard
* Cách làm:
* Để kiểm tra số lượng kRAM trên mainboard t sử dụng ngắt BIOS

int 11h

* Sau khi đưa được giá trị của byte có địa chỉ 0:410h vào AL thì ta tách bit thứ 5 và 6 của thanh ghi AL và tiến hành kiểm tra theo trình tự sau:
* Dịch phải 2 bit để so sánh (Lúc này, AL = 00000011b / b:binary)
* So sánh AL với 0

+ Nếu bằng nhau (=0) tức máy tính ko có kRAM nào và kết thúc chương trình

+ Nếu khác nhau (khác 0) tức là máy tính có kRAM, tiếp tục so sánh với giá trị khác

* So sánh AL với 1

+, Nếu bằng nhau (=1) tức là máy tính có 16 kRam, kết thúc chương trình

+. Nếu khác nhau (khác 1) thì tiếp tục so sánh với gía trị khác

* So sánh AL với 2 (dạng binary, tức so sánh với 10b)

+ Nếu bằng nhau (=2) tức là máy tính có 32 kRam, kết thúc ct

+ Nếu khác nhau (khác 2) thì suy ra máy tính có 64 kRam, kết thúc chương trình

* Các MACRO và chương trình con sử dụng:
  + HienString: hiện một xâu ký tự kết thúc bằng ‘$’ ra màn hình
  + clrscr: Xóa màn hình

# LẬP TRÌNH – CÀI ĐẶT

## Chương trình chính

#include <stdio.h>

#include <conio.h>

extern GETTIME();

extern DAOCHUOI();

extern KRAM();

void main()

{

char tl;

int cn; /\*chuc nang\*/

clrscr();

printf("\n\n Truong Dai hoc Mo Ha Noi");

printf("\n KHOA CONG NGHE THONG TIN");

printf("\n -----------o0o----------");

printf("\n\n\n BAI TAP LON MON LTHT");

printf("\n\n\n Cac SV thuc hien:");

printf("\n 1. Tran Ba Phuc Lop: 17A3");

printf("\n 2. Nguyen Thu Phuong Lop: 18A4");

printf("\n 3. Nguyen Ly Quang Lop: 17A4");

printf("\n\n\n Co tiep tuc CT (c/k)?");

tl=getch();

if(tl=='c')

{

while(1){

clrscr();

printf("\n\n CAC CHUC NANG CUA BTL");

printf("\n ---------\*\*\*----------");

printf("\n\n 1. Chuc nang GET TIME");

printf("\n 2. Vao mot xau va hien nguoc lai");

printf("\n 3. So luong kRAM then mainboard");

printf("\n 4. Thoat ve DOS");

printf("\n\n Hay chon: ");

while(1){

char term;

if(scanf("%d%c", &cn, &term) != 2 || term != '\n'){

printf("\n Nhap khong hop le!");

printf("\n\n Hay chon lai: ");

fflush(stdin);

}else{

if(cn>0&&cn<5){

break;

}else{

printf("\n Nhap khong hop le!");

printf("\n\n Hay chon lai: ");

}

}

}

switch(cn)

{

case 1: GETTIME(); break;

case 2: DAOCHUOI(); break;

case 3: KRAM(); break;

case 4: printf("\n THOAT VE DOS!"); return;

default: printf("\n Nhap khong hop le!");

}

}

}

}

## GETTIME

INCLUDE lib1.asm

.MODEL small

.DATA

header db 13,10,' CHUC NANG GET TIME'

db 13,10,' ------------------'

db 13,10,' Current time is: $'

hc db ':$'

dc db '.$'

thoat db 13,10,' An phim bat ky de ve Main menu$'

.CODE

PUBLIC \_GETTIME

\_GETTIME PROC

CLRSCR

HienString header ;Hien thong noi dung dau tien

mov ah,2ch ;Chuc nang lay gio phutvaf giay cua may tinh

int 21h

mov al,ch ;Dua gio tu ch -> al

xor ah,ah ;ah=0

call HIEN\_SO\_N ;Hien gio

HienString hc ;Hien dau ':'

mov al,cl ;Dua phut tu cl -> al

xor ah,ah ;ah=0

call HIEN\_SO\_N ;Hien phut

HienString hc ;Hien dau ':'

mov al,dh ;Dua giay tu dh -> al

xor ah,ah ;ah=0

call HIEN\_SO\_N ;Hien giay

HienString dc ;Hien dau '.'

mov al,dl ;Dua % giay tu dl -> al

xor ah,ah ;ah=0

call HIEN\_SO\_N ;Hien % giay

HienString thoat

mov ah,1 ;Cho an 1 phim

int 21h

ret

\_GETTIME ENDP

INCLUDE lib2.asm

END

## Vào một xâu và hiện ngược lại

INCLUDE lib1.asm

.MODEL small

.DATA

header db 13,10,' VAO XAU VA HIEN NGUOC LAI'

db 13,10,' -------------------------'

db 13,10,' Hay vao xau ky tu (kt bang Ctrl\_Z): $'

str db 100,0,100 dup(' ')

value db 13,10,' Dao xau vua vao la: $'

thoat db 13,10,' An phim bat ky de ve Main Menu$'

.CODE

PUBLIC \_DAOCHUOI

\_DAOCHUOI PROC

CLRSCR

HienString header

mov ah,0ah

lea dx,str

int 21h

mov cx,0

mov cl, str[1]

HienString value

lea dx,str

add dx,2

mov di,dx

add di,cx

dec di

mov ah,02

l1:

mov dl,[di]

int 21h

dec di

loop l1

HienString thoat

mov ah,1

int 21h

ret

\_DAOCHUOI ENDP

END

## Kiểm tra số lượng kRAM trên mainboard

INCLUDE lib1.asm

.MODEL small

.STACK 100h

.DATA

m1 db 13,10,' CHUC NANG SO LUONG kRAM TREN MAINBOARD'

db 13,10,' -------------------\*------------------'

db 13,10,13,10,' So luong kRAM co tren MainBoard la: $'

thoat db 13,10,' Nhan phim bat ki de thoat$'

kR0 db '0 k$'

kR16 db '16 k$'

kR32 db '32 k$'

kR64 db '64 k$'

.CODE

PUBLIC \_SOLUONGkRAM

\_SOLUONGkRAM PROC

CLRSCR ; xoa man hinh

HienString m1 ; hien noi dung m1

int 11h ; 0:410h -> AL

and al,00001100b ; tach 2 bit co thong tin lien quan den kRAM

shr al,2 ; dich bit sang phai 2 lan

jnz L\_kRAM16 ;neu khac 0 thi nhay sang L\_kRAM16

HienString kR0 ;con ko thi hien kR0

jmp Exit\_kRAM ; nhay den nhan Exit\_kRAM

L\_kRAM16:

cmp al,1 ; so sanh AL voi 1

jne L\_kRAM32 ; neu khong bang -> nhay sang nhan L\_kRam32

HienString kR16 ; neu bang thi hien xau kR16

jmp Exit\_kRAM ;nhay den nhan Exit\_kRAM

L\_kRAM32:

cmp al,2 ; so sanh AL voi 2

jne L\_kRAM64 ; neu khong bang -> nhay den nhan L\_kRam64

HienString kR32 ; neu bang, hien xau kR32

jmp Exit\_kRAM ; nhay den nhan Exit\_kRAM

L\_kRAM64:

HienString kR64 ; hien xau kR64

Exit\_kRAM:

HienString thoat

mov ah,1 ;|

int 21h ;| cho 1 ki tu tu ban phim

ret ; quay ve chuong trinh da goi no tu ct con

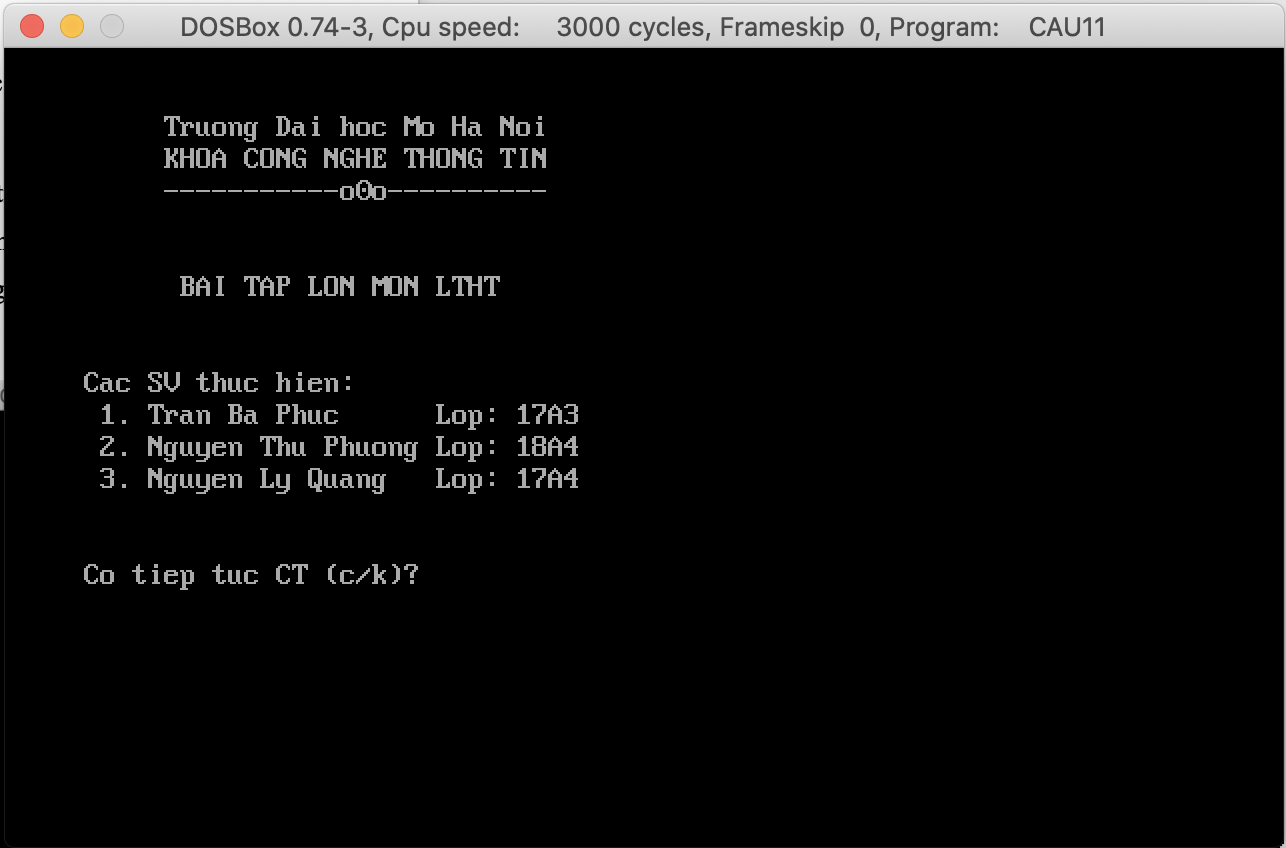
\_SOLUONGkRAM ENDP

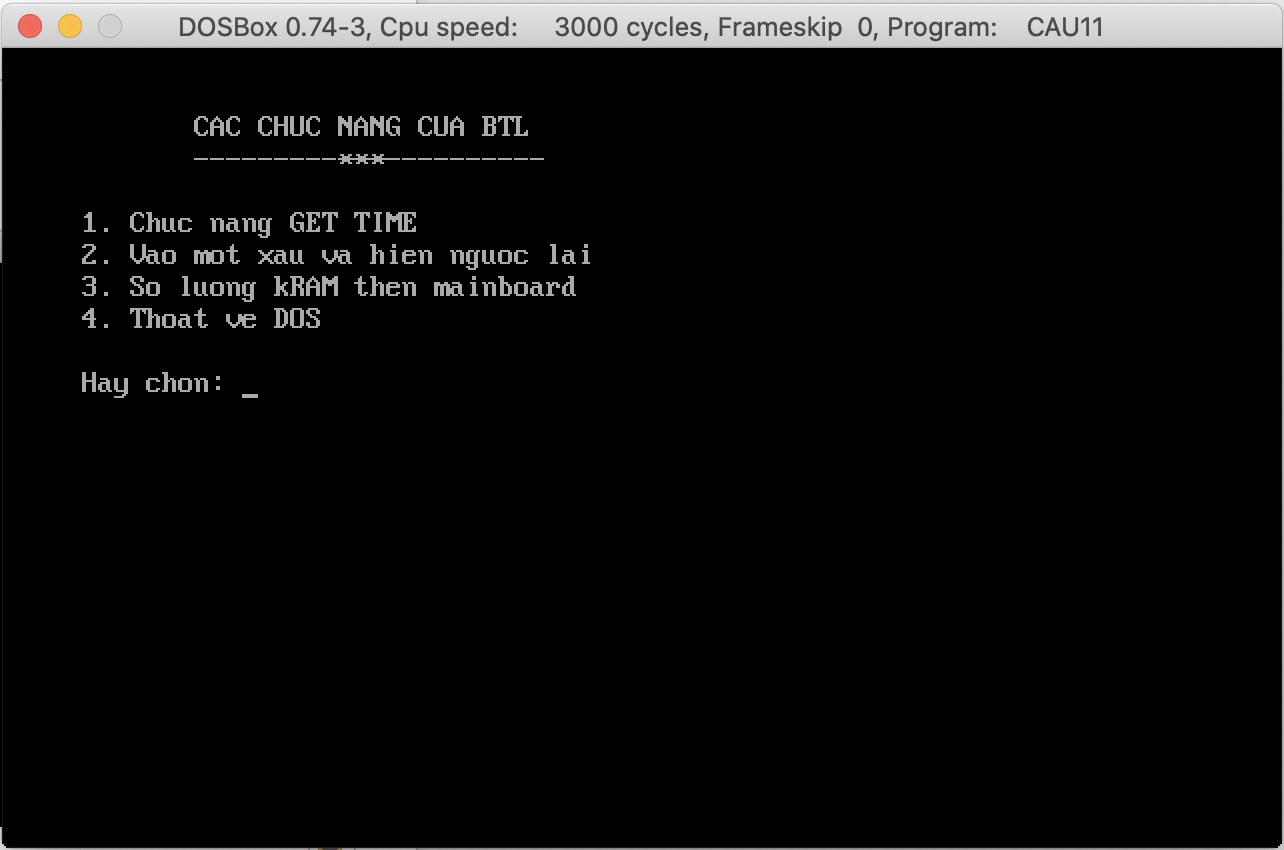
INCLUDE lib2.asm

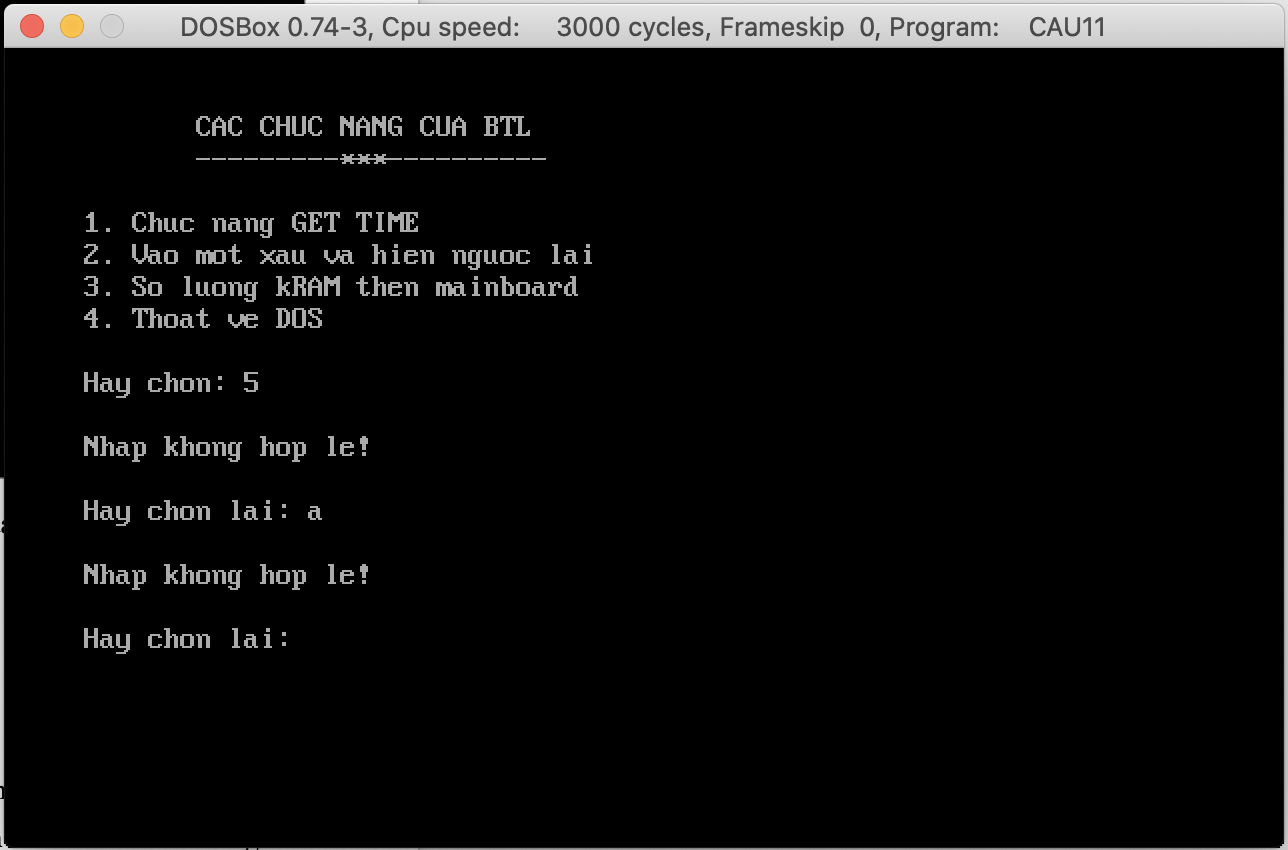
END

# DEMO TOÀN BỘ CHƯƠNG TRÌNH

## MENU chương trình

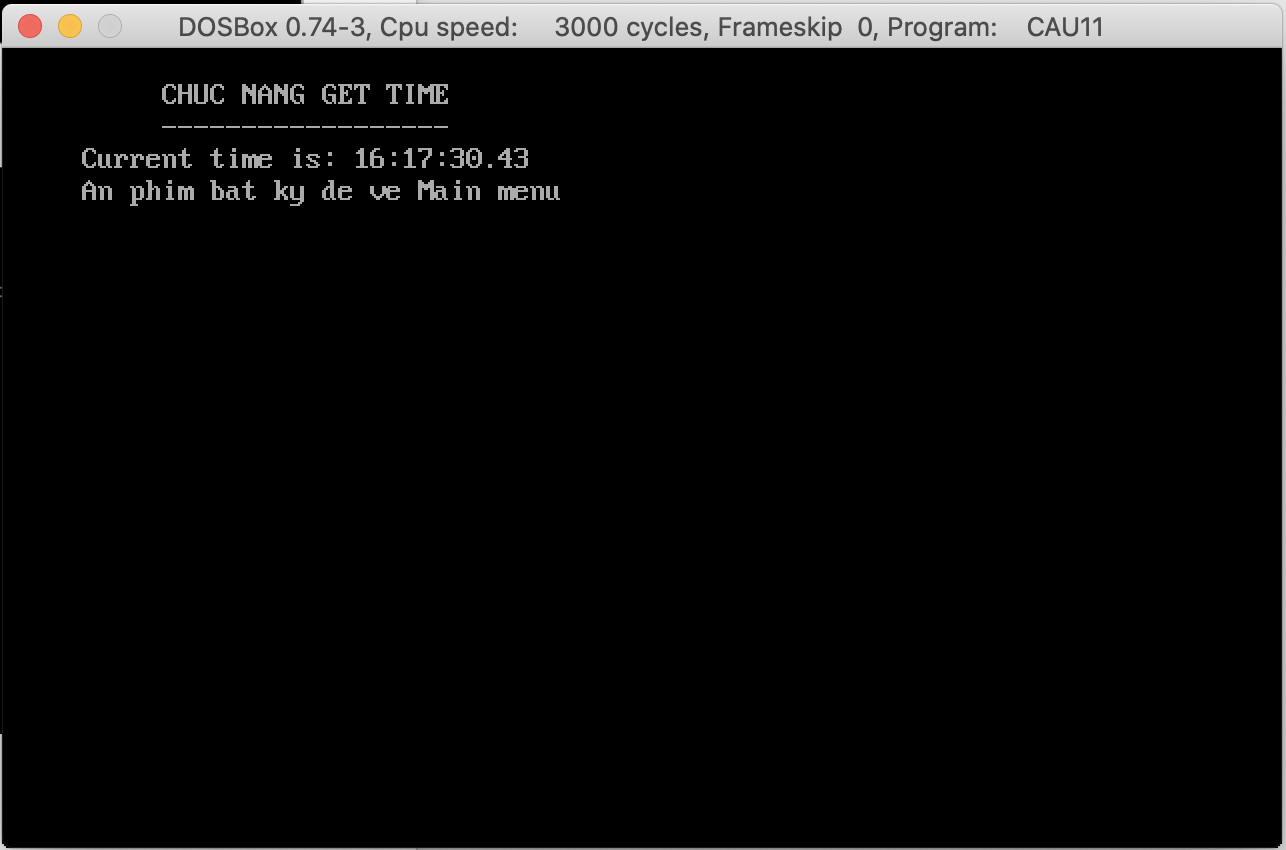




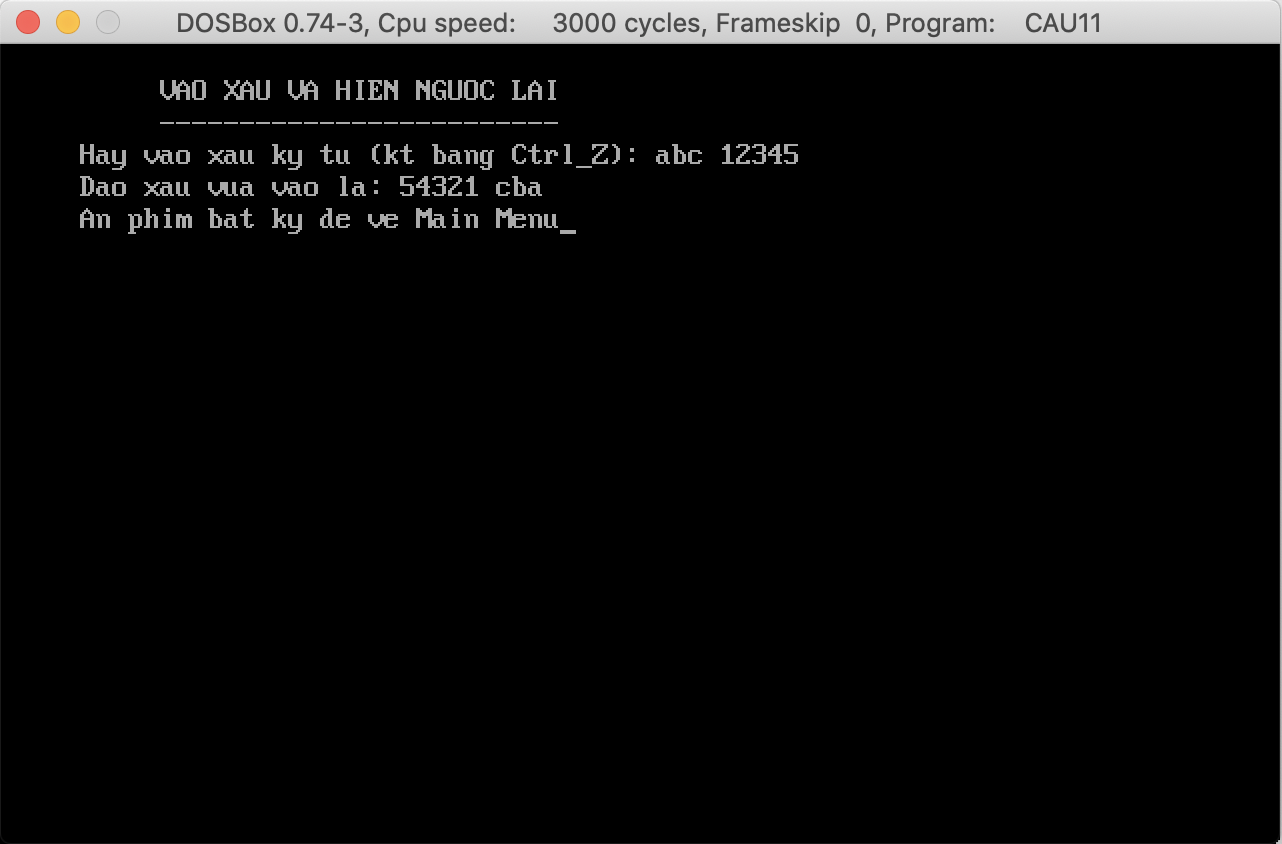




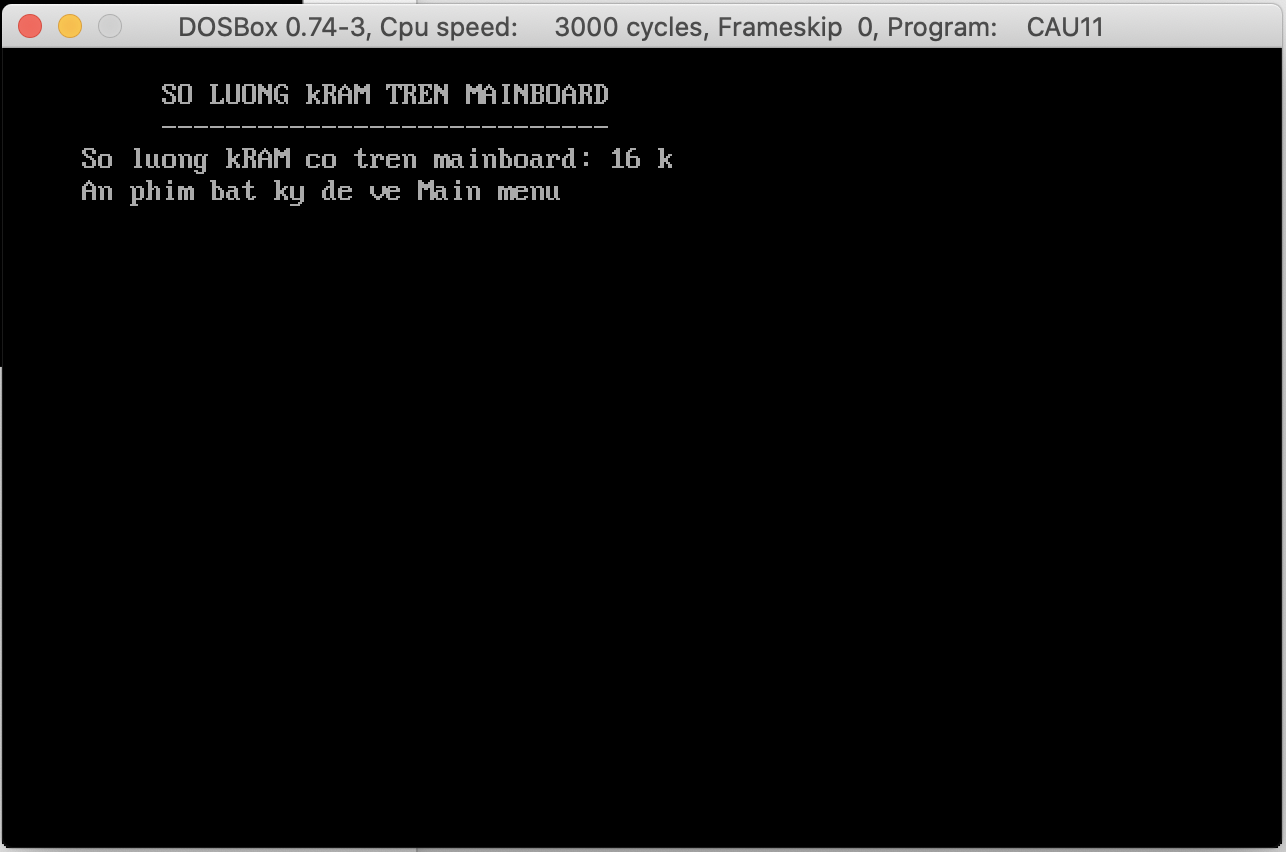
## GETTIME



## Vào một xâu và hiện ngược lại



## Kiểm tra số lượng kRAM trên mainboard



# KẾT LUẬN

Kết quả bài tập thực hiện đúng, đủ các yêu cầu được nêu. Mã nguồn ngắn gọn, dễ hiểu, thực hành đúng theo những kiến thức đã được dạy và truyền đạt. Chi tiết, tỉ mỉ, có kiểm tra dữ liệu đầu vào và cảnh báo người dùng. Giao diện đơn giản, hiệu quả, không cầu kì, rườm rà gây rối mắt.

# TÀI LIỆU THAM KHẢO

1. Giáo trình môn Lập trình hệ thống – thầy Đặng Thành Phu, khoa Công nghệ thông tin – Trường Đại học Mở Hà Nội
2. Các slide bài giảng trên lớp – môn Lập trình hệ thống, khoa Công nghệ thông tin – Trường Đại học Mở Hà Nội
3. Công cụ hỗ trợ THELP – thầy Đặng Thành Phu, khoa Công nghệ thông tin – Trường Đại học Mở Hà Nội