

BÁO CÁO ĐỒ ÁN 2 – THƯ VIỆN TIME

1. THÀNH VIÊN NHÓM

MSSV **Họ tên**

1612880	Hoàng Thiên Nữ
1612408	Phạm Khoa Nam
1612013	Lê Hoàng Anh

2. MÔ TẢ CÁCH CÀI ĐẶT MỘT SỐ HÀM QUAN TRỌNG

```
int String_to_number(char* s, int l, int r, bool check)
```

Mô tả: Hàm sẽ chuyển đổi chuỗi từ vị trí l đến vị trí r thành số. Nếu chuyển được thì trả về TRUE, ngược lại trả về FALSE. Địa chỉ giá trị số sẽ lưu vào \$v0, và biến check lưu vào \$v1

Cách cài đặt:

- Gán địa chỉ lần lượt của s, l, r vào các thanh ghi \$a0, \$a1, \$a2
- Dùng vòng lặp, chuyển từng kí tự trong chuỗi s thành số theo công thức:
 $x = s[i] - '0'$, Với x là giá trị số của kí tự thứ i trong chuỗi s
- Kiểm tra xem x có phải là số không. Nếu không phải gán giá trị mà thanh ghi \$v1 lưu là 0. Nếu phải tiếp tục vòng lặp
- Kết thúc chương trình: lưu địa chỉ giá trị số được chuyển đổi vào \$v0, gán giá trị mà thanh ghi \$v1 lưu là 1.

```
char* Date(int day, int month, int year)
```

Mô tả: Chuyển đổi thành chuỗi TIME có định dạng dd/mm/yyyy

Cách cài đặt:

- Chuyển từng giá trị số day, month, year thành chuỗi. Thuật toán chuyển đổi số thành chuỗi bằng cách lấy từng chữ số trong giá trị số để chuyển đổi thành các kí tự.
- Gán từng kí tự vào đúng vị trí của chúng qua lệnh sw
- Trả về địa chỉ của chuỗi là \$v0

char Input()*

Mô tả: Đây là hàm thực hiện yêu cầu người dùng nhập vào ngày, tháng, năm (dưới dạng chuỗi), đồng thời kiểm tra tính hợp lệ ngày, tháng, năm và trả về char* TIME (dd/mm/yyyy)

Cách cài đặt:

- Hàm sẽ cho phép nhập từng chuỗi ngày (day), tháng (month), năm (year) bằng lệnh syscall 4. Với mỗi phần nhập, hàm sẽ kiểm tra xem các chuỗi nhập vào chỉ bao gồm các kí tự số không nhờ lời gọi hàm bool String_to_number(char* s, int l, int r, bool check).
- Nếu chuỗi nhập vào hợp lệ (có thể chuyển thành số), lưu giá trị số vào stack
- Sau khi đã chuyển các chuỗi thành số và lưu giá trị vào stack, kiểm tra tính hợp lệ ngày tháng năm. Nếu không thỏa thì thực hiện lại quá trình nhập.
- Nếu thỏa, gọi hàm char* Date(int day, int month, int year) để chuyển thành chuỗi TIME có định dạng dd/mm/yyyy.
- Hàm trả về địa chỉ của chuỗi qua thanh ghi \$v0

char Convert(char* TIME, char TYPE)*

Mô tả: Hàm trả về chuỗi là một định dạng khác của chuỗi TIME. Có các định dạng sau:

- A. MM/DD/YYYY
- B. Month DD, YYYY
- C. DD Month, YYYY

Cách cài đặt:

- Ý tưởng chung: tách riêng thành các chuỗi ngày (day), tháng (month), năm (year).
- Nếu dạng nào cần chuyển số thành chuỗi thì chuyển.
- Cộng các chuỗi lại bằng cách gán giá trị từng vị trí vào chuỗi kết quả.
- Địa chỉ của chuỗi kết quả được lưu bởi thanh ghi \$v0

int Day(char TIME)*

Mô tả: Hàm cho phép lấy giá trị ngày trong chuỗi TIME

Cách cài đặt:

- Địa chỉ chuỗi TIME được lưu bởi thanh ghi \$a0
- Tìm vị trí đầu và cuối của ngày được lưu trong chuỗi TIME
- Gọi hàm String_to_number để thực hiện lấy giá trị.
- Lưu kết quả trả về vào thanh ghi \$v0

Tương tự đối với hàm Month và Year

int LeapYear(char TIME)*

Mô tả: Hàm trả về giá trị 1 nếu năm trong chuỗi TIME là năm nhuận, ngược lại 0

Cách cài đặt:

- Địa chỉ của chuỗi TIME được lưu bởi thanh ghi \$a0
- Gọi hàm Year để lấy giá trị năm trong chuỗi TIME và lưu vào stack
- Gọi hàm isLeapYear(int year) để kiểm tra năm có phải là năm nhuận không.
- Kết quả trả về được lưu bởi thanh ghi \$v0

int GetTime(char TIME_1, char* TIME_2)*

Mô tả: Hàm trả về khoảng cách năm giữa hai khoảng thời gian TIME_1 và TIME_2 (giá trị đầu vào phải thỏa điều kiện TIME_1 < TIME_2)

Cách cài đặt:

- Địa chỉ của 2 chuỗi TIME_1 và TIME_2 được lưu lần lượt vào các thanh ghi \$a0, \$a1
- Lấy giá trị năm của TIME_2 – giá trị năm của TIME_1, lưu giá trị đó vào stack
- Nếu ngày, tháng của TIME_2 nhỏ hơn ngày, tháng TIME_1 thì lấy giá trị trong stack trừ đi 1
- Địa chỉ kết quả trả về được lưu vào thanh ghi \$v0

char Weekday(char* TIME)*

Mô tả: Trả về giá trị là thứ của TIME

Cách cài đặt:

- Áp dụng công thức xác định thứ:

$$f = D + \left\lfloor \frac{13 * M - 1}{5} \right\rfloor + Y + \left\lfloor \frac{Y}{4} \right\rfloor + \left\lfloor \frac{C}{4} \right\rfloor - 2 * C$$

Với:

- D là ngày của tháng
 - M là giá trị của tháng được tính từ tháng 3 tức là Tháng 3 là 1, Tháng 4 là 2, ..., tháng 1 là 11 và tháng 2 là 12.
 - Y là 2 chữ số cuối của năm (được tính theo tháng 3 là tháng bắt đầu của năm).
Tức là nếu tháng trong chuỗi là tháng 1 và 2 thì phải trừ năm trong chuỗi đi 1.
 - C là 2 chữ số đầu của năm.
- Dùng giá trị $f \bmod 4$ để xác định thứ. Nếu phần dư của f cho 4 bằng 0 là Chủ nhật, 1 là thứ 2, ...
- Địa chỉ chuỗi thứ được trả về tại thanh ghi \$v0

3. QUI TẮC KHI VIẾT VÀ GỌI HÀM TRONG MIPS

a. Trình tự khi viết và gọi hàm

1. Lưu trữ các biến vào nơi mà hàm có thể truy cập đến (truyền tham số cho hàm)
2. Gọi hàm bằng lên **"jal"** (bắt đầu lời gọi hàm)
3. Cấu trúc của một hàm (thủ tục)

Mở đầu

- entry_label: #Tên nhãn của hàm
- addi \$sp, \$sp, -framesize #Khai báo kích thước stack
- sw \$ra, framesize - 4 (\$sp) #Lưu địa chỉ \$ra vào ngăn xếp (trường hợp có gọi hàm lồng)
- lưu tạm các thanh ghi khác (nếu cần)

Phần thân:

- Các câu lệnh thực thi

Cuối:

- Phục hồi các thanh ghi khác (nếu cần)
- lw \$ra, framesize - 4 (\$sp) #Lấy địa chỉ trở về \$ra
- addi \$sp, \$sp, framesize
- jr \$ra #Quay trở về lúc gọi hàm

Với mỗi lần gọi hàm bằng “jal”, địa chỉ lúc gọi hàm sẽ lưu vào thanh ghi \$ra. Do vậy, để tránh việc mất địa chỉ thanh ghi (khi chúng ta gọi nhiều hàm lồng nhau), cần lưu trữ \$ra vào stack. Đồng thời cần lưu 1 số thanh ghi khác: như thanh ghi cho các tham số, hay thanh ghi tạm giữ giá trị kết quả trả về (nếu cần). Vì khi gọi một hàm khác chúng ta có thể làm thay đổi giá trị của thanh ghi đó.

b. Một số nguyên tắc sử dụng thanh ghi

\$a0 - \$a3	Thanh ghi để lưu các tham số của hàm, có thể thay đổi được. Nên lưu các thanh ghi này vào stack vì thanh ghi có thể thay đổi được
\$v0 - \$v1	Thanh ghi giữ giá trị trả về của hàm
\$t0 - \$t9	Thanh ghi lưu giá trị tạm nên có thể thay đổi bất cứ lúc nào. Thủ tục gọi cần lưu lại giá trị nếu nó cần sau các lời gọi thủ tục.
\$s0 - \$s7	Thanh ghi lưu trữ. Khôi phục nếu thay đổi. Rất quan trọng, vì nếu thủ tục được gọi thay đổi các thanh ghi này thì nó phải phục hồi các thanh ghi trước khi kết thúc.
\$ra	Thanh ghi dùng để lưu vị trí của lời gọi hàm và có thể thay đổi được. Thủ tục gọi lưu lại thanh ghi này vào ngăn xếp nếu cần
\$sp	Thanh ghi con trỏ ngăn xếp phải có giá trị không đổi trước và sau khi gọi lệnh jal, nếu không thủ tục gọi sẽ không quay về được. Nên cần được khôi phục nếu thay đổi.