



BỘ MÔN KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

BÁO CÁO ĐỒ ÁN THƯ VIỆN TIME

- Giáo viên lý thuyết:
 - Phạm Tuấn Sơn
- Giáo viên thực hành:
 - Nguyễn Thanh Quân
 - Lê Viết Long

1 Thông tin chung

1.1 Thông tin nhóm

MSSV	Họ Tên	Email
1712273	Lâm Đức Anh	1712273@student.hcmus.edu.vn
1712838	Trương Khắc Triệu	1712838@student.hcmus.edu.vn
1712304	Hoàng Đức Công	1712304@student.hcmus.edu.vn

1.2 Yêu cầu đề án

- Viết chương trình minh họa có giao diện menu như sau bằng hợp ngữ MIPS:

- Nhập ngày DAY:
- Nhập tháng MONTH:
- Nhập năm YEAR:

-----Bạn hãy chọn 1 trong các thao tác dưới đây -----

- Xuất chuỗi TIME theo định dạng DD/MM/YYYY
- Chuyển đổi chuỗi TIME thành một trong các định dạng sau:
 - MM/DD/YYYY
 - Month DD, YYYY
 - DD Month, YYYY
- Cho biết ngày vừa nhập là ngày thứ mấy trong tuần:
- Kiểm tra năm trong chuỗi TIME có phải là năm nhuận không
- Cho biết khoảng thời gian giữa chuỗi TIME_1 và TIME_2
- Cho biết 2 năm nhuận gần nhất với năm trong chuỗi time (Chú ý: Hàm này phải xử dụng lại hàm ở câu 4.)
- Kiểm tra bộ dữ liệu đầu vào khi nhập, nếu dữ liệu không hợp lệ thì yêu cầu người dùng nhập lại. (Ví dụ 30/02/2012 hoặc a/1/2000 là không hợp lệ). Chú ý: năm nhuận thì tháng 2 có 29 ngày, sinh viên cần kiểm tra kỹ.

- Lựa chọn:
- Kết quả:

1.3 Bảng phân công công việc

MSSV	Họ Tên	Công việc
1712273	Lâm Đức Anh	<ul style="list-style-type: none">• Nhập dữ liệu ngày, tháng, năm vào chuỗi Time.• Kiểm tra tính hợp lệ của dữ liệu nhập.
1712838	Trương Khắc Triệu	<ul style="list-style-type: none">• Lựa chọn 1, 3, 4 trong menu trên
1712304	Hoàng Đức Công	<ul style="list-style-type: none">• Lựa chọn 2, 5, 6 trong menu trên

Lựa chọn 7 không có trong menu, thay vào đó, mỗi khi người dùng nhập ngày tháng năm vào thì chương trình tự động kiểm tra tính hợp lệ của chuỗi vừa nhập vào. (Ngoài các điều kiện thông thường khác, có thêm kiểm tra điều kiện năm ≥ 1900)

2 Môi trường lập trình và nội dung

2.1 Môi trường lập trình

Môi trường lập trình: Phần mềm Mars4_5

Ngôn ngữ lập trình: MIPS

2.2 Các hàm chính được cài đặt trong chương trình

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

- **Mô tả:**
 - Tạo chuỗi TIME theo định dạng mặc định DD/MM/YYYY
- **Tham số:**
 - day, month, year: giá trị ngày, tháng, năm do người dùng nhập vào
- **Giá trị trả về:**
 - Trả về địa chỉ chuỗi TIME đang giữ
- **Cách thức cài đặt:**
 - Lưu các \$a0, \$a1, \$a2, \$a3 vào stack (với \$a0 – day, \$a1 – month, \$a2 – year, \$a3 – chứa địa chỉ chuỗi TIME)
 - Với \$a0, \$a0 chia cho 10, lấy thương, số dư cộng cho 48 rồi ghi giá trị lần lượt vào 2 vị trí đầu của TIME(\$a3), và thêm giá trị 47 (dấu /) vào vị trí tiếp theo.
 - Tương tự với \$a1, riêng \$a2 chia lần lượt cho 1000, 100, 10, 1.
- **Cách thức gọi hàm:**

jal Date

Địa chỉ chuỗi TIME sẽ được lưu trong biến \$v0

char* Convert(char* TIME, char type)

- **Mô tả:**
 - Chuyển đổi kiểu định dạng của chuỗi TIME
- **Tham số:**
 - TIME: trỏ đến vùng nhớ lưu giá trị chuỗi ngày tháng năm cần định dạng
 - type: kiểu định muốn chuyển:
 - type = 'A': định dạng MM/DD/YYYY
 - type = 'B': định dạng Month DD, YYYY
 - type = 'C': định dạng DD Month, YYYY
- **Giá trị trả về:**
 - Lấy giá trị mà biến TIME đang giữ
- **Cách thức cài đặt:**
 - Tham số đầu vào của hàm là \$a0, \$a1 (với \$a0-địa chỉ chuỗi TIME, \$a1-ký tự type muốn chuyển)
 - Kiểm tra \$a1, nếu:
 - \$a1 = 'A': chuyển xuống label convertToTypeA, thực hiện việc hoán đổi vị trí ký tự biểu diễn ngày và tháng cho nhau, còn lại giữ nguyên.
 - \$a1 = 'B': chuyển xuống label converToTypeB, thực hiện lấy giá trị của tháng, sau đó dò để lấy chuỗi tên tháng tương ứng, thêm vào chuỗi với các vị trí theo định dạng.
 - \$a1 = 'C': tương tự như trên.
- **Cách thức gọi hàm:**

jal Convert

Địa chỉ chuỗi TIME theo định dạng mới sẽ được lưu trong biến \$v0

int Day(char* TIME)

- **Mô tả:**
 - Lấy giá trị ngày từ chuỗi TIME
- **Tham số:**
 - TIME: trỏ đến vùng nhớ lưu giá trị ngày tháng cần xử lý
- **Giá trị trả về:**
 - Trả về giá trị ngày trong chuỗi (1 --> 31)
- **Cách thức cài đặt:**
 - \$a0 chứa địa chỉ chuỗi TIME
 - Giá trị ngày tương ứng với công thức $(TIME[0] - 48) * 10 + (TIME[1] - 48)$
- **Cách thức gọi hàm**

jal Day

Giá trị trả về được lưu ở biến \$v0

int Month(char* TIME)

- **Mô tả:**
 - Lấy giá trị tháng từ chuỗi TIME
- **Tham số:**
 - TIME: trỏ đến vùng nhớ lưu giá trị ngày tháng cần xử lý
- **Giá trị trả về:**
 - Trả về giá trị ngày trong chuỗi (1 --> 12)
- **Cách thức cài đặt:**
 - \$a0 chứa địa chỉ chuỗi TIME
 - Giá trị tháng tương ứng với công thức $(TIME[3] - 48) * 10 + (TIME[4] - 48)$
- **Cách thức gọi hàm**

jal Month

Giá trị trả về được lưu ở biến \$v0

int Year(char* TIME)

- **Mô tả:**
 - Lấy giá trị năm từ chuỗi TIME
- **Tham số:**
 - TIME: trỏ đến vùng nhớ lưu giá trị ngày tháng cần xử lý
- **Giá trị trả về:**
 - Trả về giá trị ngày trong chuỗi (≥ 1900) (Luôn đúng bởi vì chương trình kiểm tra lúc người dùng nhập có năm ≥ 1900 thì mới hợp lệ)
- **Cách thức cài đặt:**
 - \$a0 chứa địa chỉ chuỗi TIME
 - Giá trị tháng tương ứng với công thức $(\text{TIME}[6] - 48) * 1000 + (\text{TIME}[7] - 48) * 100 + (\text{TIME}[8] - 48) * 10 + \text{TIME}[9] - 48$
- **Cách thức gọi hàm**

jal Year

Giá trị trả về được lưu ở biến \$v0

int LeapYear(char* TIME)

- **Mô tả:**
 - Kiểm tra năm nhuận
- **Tham số:**
 - TIME: trỏ đến vùng nhớ lưu giá trị ngày tháng cần xử lý

- **Giá trị trả về:**
 - 0: năm trong chuỗi TIME không phải là năm nhuận
 - 1: năm trong chuỗi TIME là năm nhuận
- **Cách thức cài đặt:**
 - Lấy giá trị trả về của hàm year (jal year) gán vào \$a0
 - Gán giá trị 400 cho \$t0, thực hiện phép chia \$a0 cho \$t0
 - Nếu số dư bằng thì trả về True
 - Ngược lại gán \$t0 bằng 4 , thực hiện chia \$a0 cho \$t0 , nếu số dư khác không thì trả về False, ngược lại tiếp tục gán \$t0 bằng 100 và thực hiện kiểm tra.
 - True thì gán \$v0 = 1 ngược lại gán \$v0 = 0
- **Cách thức gọi hàm**

jal check_leap_year

Giá trị trả về được lưu ở biến \$v0

```
int GetTime(char* TIME_1, char* TIME_2)
```

- **Mô tả:**
 - Tính thời gian cách biệt giữa giá trị năm của chuỗi TIME_1 và TIME_2
- **Tham số:**
 - TIME_1, TIME_2: trỏ đến vùng nhớ lưu giá trị ngày tháng xử lý
- **Giá trị trả về:**
 - Số năm cách biệt (≥ 0)
- **Cách thức cài đặt:**
 - Chắc chắn chuỗi TIME_1 luôn luôn sau chuỗi TIME_2 về thời gian(bằng cách kiểm tra và nếu chuỗi TIME_1 bé hơn thì hoán đổi 2 chuỗi)
 - Lấy giá trị hai năm của 2 chuỗi trừ nhau, nếu tháng của chuỗi TIME_1 bé hơn của TIME_2 thì giá trị trên trừ thêm 1.

- Nếu tháng của chuỗi TIME_1 bằng của TIME_2 nhưng ngày bé hơn thì giá trị trả về trừ thêm 1.
- Giá trị trả về lưu trong \$v0.
- Cách thức gọi hàm
 jal GetTime
 Giá trị trả về được lưu ở biến \$v0

char* Weekday(char* TIME)

- Mô tả:
 - Cho biết giá trị ngày trong chuỗi TIME là thứ mấy trong tuần
- Tham số:
 - TIME: trỏ đến vùng nhớ lưu giá trị ngày tháng xử lý
- Giá trị trả về:
 - Trả về thứ trong tuần thuộc tập giá trị sau: {Mon; Tues; Wed; Thurs; Fri; Sat; Sun}
- Cách thức cài đặt:
 - Lần lượt giá trị về ngày, tháng, năm (số nguyên) trong chuỗi TIME, lưu vào stack.
 - Áp dụng công thức có sẵn về tính thứ trong tuần. Công thức lấy từ <https://nguyenvanhieu.vn/cach-tinh-thu-khi-biet-ngay-thang-nam/>

```
JMD = (day + ((153 * (month + 12 * ((14 - month) / 12) - 3) + 2) / 5) +
        (365 * (year + 4800 - ((14 - month) / 12))) +
        ((year + 4800 - ((14 - month) / 12)) / 4) -
        ((year + 4800 - ((14 - month) / 12)) / 100) +
        ((year + 4800 - ((14 - month) / 12)) / 400) - 32045) % 7;
```

- Có những giá trị được sử dụng lại được lưu vào 2 biến



```
$s0 = (14 - month) / 12  
$s1 = year + 4800 - a
```

- Thực hiện tuần tự các phép tính như công thức ở trên.
- Cách thức gọi hàm

`jal week_day`

Giá trị trả về được lưu ở biến \$v0. Sau đó lưu cho biến \$a0 mang kết quả của \$v0.

- Tiếp tục gọi hàm :

`jal print_week_day`

Giá trị trả về là chuỗi mang kết quả là thứ trong tuần, lưu kết quả ở \$v0.

3 Kết quả chương trình

3.1 Nhập dữ liệu từ người dùng

- Khi chạy chương trình, lần lượt xuất hiện

```
Nhap ngay DAY: 21  
Nhap thang MONTH: 01  
Nhap nam YEAR: 1999
```

3.2 Thực hiện các chức năng trong menu

3.2.1 Chức năng 1: Xuất chuỗi TIME theo định dạng DD/MM/YYYY

```
Hay nhap lua chon cua ban: 1  
21/01/1999
```

3.2.2 Chức năng 2: Xuất chuỗi TIME theo định dạng

3.2.2.1 Định dạng 'A': MM/DD/YYYY

```
A. MM/DD/YYYY  
B. Month DD, YYYY  
C. DD Month, YYYY  
Chon dinh dang: A  
01/21/1999
```

3.2.2.2 Định dạng 'B': Month DD, YYYY

```
Hay nhap lua chon cua ban: 2  
A. MM/DD/YYYY  
B. Month DD, YYYY  
C. DD Month, YYYY  
Chon dinh dang: B  
January 21, 1999
```

3.2.2.3 Định dạng 'C': DD Month, YYYY

```
Hay nhap lua chon cua ban: 2  
A. MM/DD/YYYY  
B. Month DD, YYYY  
C. DD Month, YYYY  
Chon dinh dang: C  
21 January, 1999
```

3.2.3 Chức năng 3: Cho biết ngày vừa nhập là ngày thứ mấy trong tuần

- Chuỗi TIME chứa ngày 07/05/2019 -> Thứ ba trong tuần

```
Hay nhap lua chon cua ban: 3  
Tuesday
```

3.2.4 Chức năng 4: Kiểm tra năm trong chuỗi TIME có phải năm nhuận không?

- Chuỗi TIME chứa ngày 21/01/1999 -> 1999 không phải năm nhuận

```
Hay nhap lua chon cua ban: 4  
Khong la nam nhuan
```

3.2.5 Chức năng 5: Cho biết khoảng cách năm giữa 2 chuỗi TIME_1 và TIME_2

- Chuỗi TIME_1 chứa 07/05/2019
- Chuỗi TIME_2 chứa 11/12/2025

```
Hay nhap lua chon cua ban: 5  
Nhap ngay DAY: 11  
Nhap thang MONTH: 12  
Nhap nam YEAR: 2025  
Khoang cach giua 2 ngay tren:  
6
```

3.2.6 Cho biết 2 năm nhuận gần nhất với năm trong chuỗi TIME

- Chuỗi TIME chứa ngày 21/01/1999 -> kết quả 2 năm nhuận gần năm 1999 nhất

```
2 nam nhuan gan nhat :  
1996, 2000
```

4 Các tài liệu đã tham khảo

Slide bài giảng – Moodle bộ môn kiến trúc máy tính và hợp ngữ trường ĐH KHTN

<https://vietcodes.github.io/algo/mips>

<https://nguyenvanhieu.vn/cach-tinh-thu-khi-biet-ngay-thang-nam/>

5 Đánh giá công việc

Tên chức năng	Mức độ hoàn thành	Ghi chú
Nhập xuất , kiểm tra hợp lệ dữ liệu	100%	Hoạt động tốt
Chức năng 1	100%	Hoạt động tốt
Chức năng 2	100%	Hoạt động tốt
Chức năng 3	100%	Hoạt động tốt
Chức năng 4	100%	Hoạt động tốt
Chức năng 5	100%	Hoạt động tốt
Chức năng 6	100%	Hoạt động tốt