



BÁO CÁO ĐỒ ÁN THỰC HÀNH 2:

THƯ VIỆN TIME

Môn: Kiến trúc máy tính và Hợp ngữ

Sinh viên thực hiện:

MSSV	Họ tên
1712645	Bùi Thị Cẩm Nhung
1712747	Nguyễn Ngọc Băng Tâm
1712856	Huỳnh Văn Tú

TPHCM, 2019



Mục lục

I. Giới thiệu	3
1.1. Mô tả đề án	3
1.2. Đánh giá mức độ hoàn thành	3
1.3. Phân công công việc	3
1.4. Giao diện chương trình	4
II. Nội dung thực hiện	5
2.1. Quy tắc viết và gọi hàm	5
2.1.1 Quy tắc comment:	5
2.1.2 Sử dụng thanh ghi:	5
2.1.3. Quy tắc gọi thủ tục:	5
2.2. Mô tả cài đặt các hàm sử dụng	6
2.2.1. Kiểm tra tính hợp lệ của dữ liệu	6
2.2.2. Kiểm tra năm nhuận	6
2.2.3. Chuyển đổi ngày, tháng, năm dưới dạng số về chuỗi	7
2.2.4. Chuyển đổi kiểu định dạng của chuỗi TIME:	8
2.2.5. Hàm lấy giá trị (số) ngày, tháng, năm	9
2.2.6. Cho biết hai năm nhuận gần nhất	10
2.2.7. Tìm thứ trong tuần	11
2.2.8. Tính khoảng cách giữa hai ngày, tháng, năm bất kỳ	11
III. Thử nghiệm	12
IV. Tài liệu tham khảo	12



I. Giới thiệu

1.1. Mô tả đồ án

Xây dựng thư viện Time trong MIPS cho phép người dùng:

- Nhập ngày, tháng, năm và kiểm tra tính hợp lệ.
- Hiển thị ngày, tháng, năm dưới nhiều định dạng khác nhau.
- Cho biết thứ trong tuần của ngày, tháng, năm đã nhập.
- Kiểm tra năm nhuận và tìm hai năm nhuận gần nhất.
- Tính khoảng cách giữa hai ngày, tháng, năm bất kỳ.

1.2. Đánh giá mức độ hoàn thành

- Đánh giá tổng thể:
- Chi tiết từng yêu cầu:

STT	Yêu cầu	Hoàn thành
1	Nhập ngày, tháng, năm	x
2	Kiểm tra tính hợp lệ của ngày, tháng, năm vừa nhập	x
3	Xuất chuỗi ngày, tháng, năm theo định dạng DD/MM/YYYY	x
4	Chuyển đổi chuỗi ngày, tháng, năm thành định dạng MM/DD/YYYY, Month DD, YYYY hay DD Month, YYYY	x
5	Cho biết ngày, tháng, năm đã nhập thuộc thứ mấy trong tuần	x
5	Kiểm tra năm vừa nhập có là năm nhuận hay không	x
6	Cho biết khoảng thời gian giữa hai chuỗi ngày, tháng, năm bất kỳ	
7	Cho biết hai năm nhuận gần nhất với năm trong chuỗi	x

1.3. Phân công công việc

MSSV	Họ tên	Công việc thực hiện
1712645	Bùi Thị Cẩm Nhung	
1712747	Nguyễn Ngọc Băng Tâm	



1712856	Huỳnh Văn Tú	
---------	--------------	--

1.4. Giao diện chương trình

Xyz



II. Nội dung thực hiện

2.1. Quy tắc viết và gọi hàm

2.1.1 Quy tắc comment:

a. Procedure header:

- Trước mỗi thủ tục nên thực hiện comment với những nội dung:

- Tên thủ tục
- Các đối số và giá trị trả về (nếu có)
- Chức năng của thủ tục
- Các thanh ghi sử dụng

b. Inline comment:

- Các dấu # nằm trên cùng một đường thẳng.
- Phần comment sử dụng tên biến thay vì thanh ghi.

2.1.2 Sử dụng thanh ghi:

- Thanh ghi \$t: không cần lưu lại khi gọi thủ tục
- Thanh ghi \$s: cần lưu vào stack tất cả thanh ghi s có nguy cơ bị nạp chồng khi gọi thủ tục.
- Thanh ghi \$ra: lưu vào stack ở đầu và pop khỏi stack ở cuối thủ tục.

2.1.3. Quy tắc gọi thủ tục:

- Cần tuân thủ quy tắc này cho tất cả thủ tục, bao gồm cả main.
- Quy ước: caller là thủ tục gọi, callee là thủ tục được gọi.

a. Trước khi caller gọi thủ tục:

- Truyền đối số vào các thanh ghi \$a0 - \$a3. Nếu có nhiều hơn 4 đối số, push các đối số còn lại vào stack.
- Gọi lệnh jal để nhảy đến callee và lưu giá trị trả về vào \$ra.

b. Khi callee được gọi:

- Push \$ra vào stack.



- Push tất cả thanh ghi \$s mà callee sẽ sử dụng vào stack.

- Giảm \$sp.

c. Sau khi callee được gọi và trước khi trả giá trị về:

- Gán kết quả trả về vào \$v0.

- Tăng \$sp.

- Phục hồi các thanh ghi \$s và \$ra.

- Trả về bằng lệnh jr \$ra.

2.2. Mô tả cài đặt các hàm sử dụng

2.2.1. Kiểm tra tính hợp lệ của dữ liệu

Hàm tương ứng: **bool isValid(int day, int month, int year)**

Mô tả:

- Các dữ liệu về ngày, tháng, năm do người dùng nhập vào sẽ được chuyển đổi từ dạng chuỗi sang dạng số. Nếu chuyển đổi không thành công, kết quả trả về là **-1**.
- Ngày, tháng, năm dưới dạng số được kiểm tra tính hợp lệ dựa trên thuật toán sau:
 - o Nếu không thể chuyển đổi một trong số dữ liệu do người dùng nhập vào dưới dạng số (tức kết quả chuyển đổi bằng **-1**), trả về không hợp lệ.
 - o Nếu giá trị ngày, tháng, năm nhỏ hơn 1, trả về không hợp lệ.
 - o Nếu giá trị tháng lớn hơn 12, trả về không hợp lệ.
 - o Gọi giá trị trả về của hàm **int maxDayInMonth(month, year)** là số lượng ngày tối đa có thể có của tháng hiện tại. So sánh giá trị ngày với giá trị trả về này. Nếu vượt quá, trả về không hợp lệ.
 - o Trả về kết quả hợp lệ nếu không rơi vào bất cứ trường hợp nào ở trên.
- Giá trị trả về: **true** nếu ngày, tháng, năm hợp lệ, ngược lại là **false**.

2.2.2. Kiểm tra năm nhuận

Hàm tương ứng: **bool isLeapYear(int year)**



Mô tả:

- Giá trị năm truyền vào được kiểm tra có phải là năm nhuận hay không theo thuật toán dưới đây:
 - o Nếu năm chia hết cho 400, trả về là năm nhuận.
 - o Nếu năm chia hết cho 100, trả về không phải là năm nhuận.
 - o Nếu năm chia hết cho 4 (kết hợp với điều kiện trên, thỏa năm vừa chia hết cho 4 nhưng không chia hết cho 100), trả về là năm nhuận.
 - o Trả về không phải là năm nhuận nếu không rơi vào bất cứ trường hợp nào ở trên.
- Giá trị trả về: **true** nếu năm đó là năm nhuận, ngược lại là **false**.

2.2.3. Chuyển đổi ngày, tháng, năm dưới dạng số về chuỗi

Hàm tương ứng: **char* date(int day, int month, int year, char* time)**

Mô tả:

- Các giá trị ngày, tháng, năm dạng số sau khi đã được kiểm tra tính hợp lệ sẽ được chuyển đổi thành dạng chuỗi và lưu dưới dạng DD/MM/YYYY vào biến time theo thuật toán như sau:
 - o Gọi i là vị trí của ký tự hiện tại trong chuỗi time. Ban đầu i = 0, tức vị trí đầu chuỗi (tương ứng với base address của time).
 - o Đưa giá trị ngày vào chuỗi:
 - Vì ta sẽ lấy từng chữ số trong ngày theo thứ tự từ hàng đơn vị trở lên nên kết quả nhận được sẽ là một chuỗi ngược. Ví dụ: giá trị ngày là 12 thì mỗi lần thực hiện, ta nhận được các ký tự lần lượt là '2', '1'. Do đó, ta sẽ tiến hành gán từng ký tự này theo chiều từ phải sang trái để cho ra kết quả đúng.
 - Chuỗi time sử dụng 2 ký tự đầu tiên để biểu diễn ngày, do đó ta gán i = 2. Với từng ký tự nhận được, ta giảm i đi 1 đơn vị và đưa vào time[i].
 - Sau khi hoàn tất, ta cộng giá trị của i lên 2 để đưa về vị trí của ký tự cuối cùng hiện có trong chuỗi time.



- Thêm ký tự '/' để phân cách ngày và tháng.
 - Thực hiện tương tự để đưa giá trị tháng và năm vào chuỗi.
 - Sau khi đã hoàn tất, đưa ký tự '\0' vào cuối để kết thúc chuỗi.
- Giá trị trả về: base address của chuỗi time.

2.2.4. Chuyển đổi kiểu định dạng của chuỗi TIME:

Hàm tương ứng: **char* Convert (char* time, char type)**

Mô tả:

- Hàm nhận kiểu định dạng do người dùng nhập vào và trả về chuỗi theo định dạng mới về thuật toán sau:
 - MM/DD/YYYY:
 - Hoán vị giá trị ngày và tháng bằng biến tạm thông qua các thanh ghi \$t.
 - Month DD, YYYY và DD Month, YYYY:
 - Thuật toán xử lý hai định dạng này gần giống nhau, chỉ khác về định dạng chuỗi con và thứ tự nối chuỗi.
 - Xem chuỗi time mới gồm 3 chuỗi con gồm chuỗi ngày, tháng và năm. Do đó ta cần xử lý định dạng 3 chuỗi con rồi thực hiện nối chuỗi.
 - *Tháng*: dùng hàm **int month(char* time)** để rút trích giá trị nguyên của tháng từ chuỗi time. Kết hợp thêm hàm **char * convertMonth(int month)** nhận giá trị tháng (kiểu nguyên) và xử lý switch – case để trả về tên tháng (kiểu chuỗi) tương ứng.
 - *Ngày và năm*: dùng vùng nhớ phụ là buffer và buffer_2 để lưu giá trị ngày. Ngoài ra còn lưu thêm các ký tự ' ' và ',' để giống định dạng yêu cầu.
 - Để lưu chuỗi mới vào vùng nhớ của chuỗi time, cần dùng hàm hỗ trợ sao chép chuỗi **void strcpy(char *&des, char *src)**. Hàm này sẽ duyệt lần lượt từ vị trí đầu chuỗi và sao chép từ chuỗi

nguồn sang chuỗi đích. Việc sao chép kết thúc khi gặp ký tự ‘\0’.

- Với định dạng Month DD, YYYY sao chép chuỗi tháng sang chuỗi time. Với định dạng DD Month, YYYY sao chép chuỗi ngày sang time.
- Để nối 2 chuỗi con còn lại vào chuỗi time, cần dùng hàm hỗ trợ nối chuỗi **void strcat(char *&des, char *src)**. Hàm này thực hiện theo 2 bước:
 - Ở nhãn findEndFirstStr, duyệt từ đầu đến cuối chuỗi đích đến khi gặp ký tự ‘\0’ thì nhảy sang nhãn appendStr.
 - Ở nhãn appendStr, thực hiện sao chép từ đầu đến cuối chuỗi nguồn và lưu vào chuỗi đích tương tự hàm sao chép.

2.2.5. Hàm lấy giá trị (số) ngày, tháng, năm

Hàm tương ứng:

- **int day(char* time)**
- **int month(char* time)**
- **int year(char* time)**

Mô tả:

- Để trích xuất từng giá trị ngày, tháng, năm được lưu trong chuỗi time có dạng DD/MM/YYYY, cho rằng đã cài đặt thành công hàm hỗ trợ **int stringToNumber(string buffer)**, ta thực hiện theo thuật toán sau:
 - Trích xuất giá trị ngày:
 - Mục đích: Đưa giá trị ngày trong chuỗi time sang chuỗi buffer và gọi hàm chuyển đổi chuỗi buffer thành số đã được cài đặt.
 - Thực hiện: Duyệt qua từng ký tự của chuỗi time cho đến khi nào gặp được ký tự ‘/’ đầu tiên, với mỗi lần lặp, thực hiện chép ký tự tương ứng của chuỗi time vào buffer.



- Trích xuất giá trị tháng:
 - Do chuỗi time được lưu dưới dạng DD/MM/YYYY, do đó để tới được vị trí bắt đầu của chuỗi tháng, ta cần bỏ qua các ký tự từ đầu cho đến dấu '/' thứ nhất.
 - Sau khi đã đến được vị trí của ký tự đầu tiên thuộc chuỗi tháng, thực hiện tương tự các bước trích xuất giá trị ngày để thu được giá trị tháng tương ứng.
- Trích xuất giá trị năm:
 - Thực hiện tương tự trích xuất giá trị tháng, tuy nhiên ta cần bỏ qua các ký tự từ đầu cho đến dấu '/' thứ hai.
 - Thực hiện chép lần lượt ký tự của chuỗi time vào buffer cho đến khi gặp ký tự kết thúc chuỗi '\0'.
- Giá trị trả về: giá trị số của ngày, tháng, năm tương ứng trong chuỗi.

2.2.6. Cho biết hai năm nhuận gần nhất

Hàm tương ứng: **void print2LYear(char *time)**

Mô tả:

- Thực hiện trích xuất giá trị năm trong chuỗi time và lưu vào biến year.
- Hai năm nhuận gần nhất với year được xác định dựa trên thuật toán sau:
 - Xác định năm đầu tiên chia hết cho 4 tính từ năm hiện tại trở về trước bằng cách:
 - Chia nguyên $year = year / 4$.
 - Gán $year = year * 4$.
 - Các năm nhuận đều mang điểm chung là chia hết cho 4. Do đó để xác định năm nhuận kế tiếp tính từ năm hiện tại, ta lấy năm đầu tiên chia hết cho 4 đã tính được ở bước trên cộng thêm cho 4.
 - Kiểm tra xem năm này có phải là năm nhuận hay không bằng hàm isLeapYear. Nếu đúng, tăng số lượng năm nhuận gần nhất đã xác định được lên 1. In năm đó ra màn hình.
 - Tiếp tục cộng thêm bước nhảy 4 năm cho đến khi nào đã in đủ 2 năm nhuận gần nhất.



2.2.7. Tìm thứ trong tuần

Hàm tương ứng: **int weekDay(char* time)**

Mô tả:

- Thực hiện trích xuất giá trị từ 0 → 6 (tương ứng với Chủ Nhật, Thứ 2, ..., Thứ 7) với tham số đầu vào là chuỗi time.
- Xác định giá trị ngày trong chuỗi time là ngày thứ mấy trong tuần theo công thức sau:

$$date = (day + 2 * month + 3 * \frac{(month+1)}{5} + year + (\frac{year}{4})) \% 7$$

2.2.8. Tính khoảng cách giữa hai ngày, tháng, năm bất kỳ

Hàm tương ứng: **int getTime(char* time_1, char* time_2)**

Mô tả:

- Để việc so sánh các giá trị ngày, tháng, năm giữa hai chuỗi time_1, time_@ được dễ dàng, ta thực hiện trích xuất các giá trị này của time_1 và time_2 và lưu vào các biến year_1, year_2, month_1, month_2, day_1, day_2.
- Xác định khoảng cách giữa hai ngày, tháng, năm bất kỳ được thực hiện theo thuật toán sau:

- o Gán $res = year_1 - year_2$.
- o Nếu $res = 0$, tức hai chuỗi có cùng giá trị năm, trả về số năm cách biệt là 0.
- o Ngược lại, mặc định giá trị năm cách biệt sẽ bằng trị tuyệt đối của hiệu hai năm $res = abs(res)$.
- o Quy ước chuỗi có năm lớn hơn là time_1, hoán đổi các biến cần thiết.

Tiến hành so sánh giá trị tháng:

- Nếu $month_2 < month_1$, tức chưa hoàn thành một năm cách biệt cuối cùng, giảm số năm cách biệt đi một đơn vị, trả về kết quả.
- Nếu $month_2 > month_1$, trả về kết quả.



- Nếu hai giá trị tháng bằng nhau, so sánh giá trị ngày:
 - Nếu $\text{day_2} < \text{day_1}$, tức chưa hoàn thành một năm cách biệt cuối cùng, giảm số năm cách biệt đi một đơn vị, trả về kết quả.
 - Ngược lại, ngay lập tức trả về kết quả.
- Kết quả trả về: số năm cách biệt giữa time_1 và time_2 .

III. Thử nghiệm

IV. Tài liệu tham khảo

[1] Assembly Language Convention Guide Brown University