

پروژه‌ی پایانی

مدرس: مصطفی نوری بایگی

زمان تحویل: ۹ بهمن ۱۳۹۸

مبانی کامپیوتر و برنامه‌سازی

نیمسال اول ۹۸-۹۹



طرح کلی پروژه

در این پروژه قصد داریم با فرمت خاصی از فایل به نام میدی (MIDI)، قطعه‌هایی از موسیقی را پخش کنیم.

میدی مخفف Musical Instrument Digital Interface و به معنای «رابط دیجیتال ابزار موسیقی» است؛ به عبارت دیگر، میدی یک استاندارد برای متصل نمودن وسایل موسیقی به کامپیوتر بوده و به نوعی زبان صحبت کردن آنها با یکدیگر است. از خصوصیات فرمت میدی تجزیه اصوات و امکان تفکیک صدای سازهای مختلف موجود در یک قطعه موسیقی است.

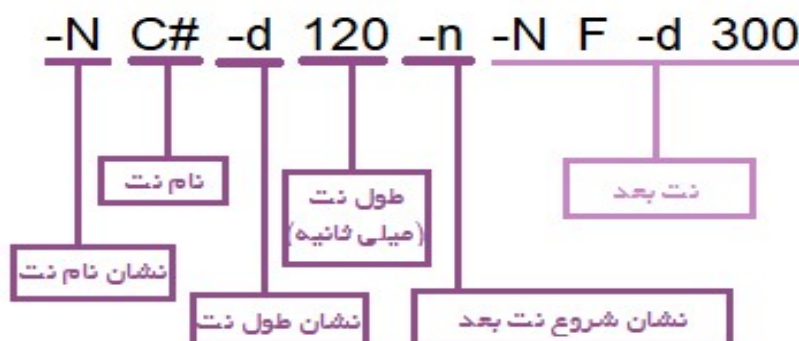
۱. توضیح اجزای پروژه

برای درک ساختار فایل‌های میدی و پیاده‌سازی فازهای ذیل، ابتدا مستنداتی را که در قسمت توضیحات تکمیلی در اختیار شما قرار داده شده است به طور کامل مطالعه کنید.

فاز اول – خواندن نت :

در این فاز با استفاده از کتابخانه‌ای که در اختیارتان قرار داده‌ایم، نت‌هایی را که در ضمیمه همین فایل (NOTES.txt) آمده است به همراه مدت زمان اجرای آن‌ها خوانده و سپس پخش کنید.

دقت کنید که در این فاز شما یک رشته دریافت خواهید کرد که شامل چند قسمت و مشابه شکل زیر است.



فاز دوم – خواندن اطلاعات فایل میدی:

در این فاز ساختار ابتدایی پروژه را پیاده سازی می کنیم. با استفاده از یک تابع مشخصات فایل میدی مورد نظر یعنی Header Chunk را که شامل Chunk Type، length و Data را –که خود شامل فرمت فایل میدی (format)، تعداد ترک ها (tracks) و نوع زمان بندی (division) می باشد– دریافت می کنیم و برای کاربر چاپ می کنیم .

برای این امر می توانید ساختاری مطابق با مقادیر موجود در header chunk ساخته و بایت های خوانده شده را در این ساختار ذخیره کنید .

برای راحتی فرایند خواندن بایت ها می توانید تابعی که در ورودی FD_ID و length (تعداد بایت های هر بخش از Header Chunk که باید خوانده شود) دریافت می کند را طراحی کنید . در زیر شکل کلی این تابع را مشاهده می کنید: (در مقدار بازگشتی تابع می توانید رخداد یا عدم رخداد خطا را مشخص کنید).

```
int read_bytes (int FD_ID , int length);
```

Header Chunk				
Chunk Type	length	Data		
4 bytes (ascii)	4 bytes (32-bit binary)	<-- length (= 6 bytes) -->		
		16-bit	16-bit	16-bit
MThd	<length>	<format>	<tracks>	<division>

فاز سوم – تشخیص نت ها :

در این فاز با استفاده از یک تابع شروع به خواندن Track Chunk ها می کنیم که تعداد آن ها را در فاز قبل ذخیره کرده ایم؛ در این تابع باید قابلیت شناسایی نت ها و event ها (مانند end of track که مشخصه پایان Track است) و سپس ذخیره آن ها بصورت آرایه ای از نت ها و event ها را پیاده سازی کنیم (این آرایه ها به عنوان ورودی به تابع داده می شوند).

فاز چهارم – پخش میدی :

در این فاز با استفاده از آرایه ای از نت ها به همراه Event ها که در فاز قبلی آن ها را شناسایی کرده ایم همانند فاز اول عمل کرده و نت ها را پخش می کنیم.

نمره امتیازی :

– Clean Code: (تا ۵ امتیاز) کد شما باید خوانا و به راحتی قابل فهم باشد، از مواردی که بر روی خوانایی کد اثر دارد، علاوه بر شکل ظاهری (مانند رعایت دندانه گذاری و رعایت بلوک بندی در دستورات شرط، حلقه و ...) نام گذاری صحیح متغیرها، کامنت گذاری و ماژول بندی صحیح کد است (یعنی کدی که به طور مستقیم در تابع main است حداقل بوده و تا حد امکان کد در قالب توابع پیاده سازی شود). لازم به ذکر است که خوانایی کد شما از الزامات پروژه بوده و در صورت نداشتن این ویژگی نمره ای متناسب از شما کسر خواهد شد!

– User Interface: (تا ۱۵ امتیاز) می توانید برای برنامه خود از یک رابط کاربری (در این مورد در صورت درخواست شما توضیحات بیشتری داده خواهد شد) ، چه در قالب متن و چه گرافیک استفاده کنید.

– پیاده سازی تابع beep: (تا ۱۰ امتیاز) در این قسمت می توانید تابع beep را که به صورت آماده در اختیارتان قرار داده ایم، خودتان پیاده سازی کنید.

– استفاده از git: (تا ۵ امتیاز) git یک کنترل کننده ورژن است و استفاده از آن می تواند به مدیریت بهتر پروژه و کد شما کمک کند. در پروژه هایی با ابعاد بزرگتر و تیمی نیز وجود git تقریباً الزامی به نظر میرسد. پس فرصت خوبی است تا نحوه کار با این ابزار پرطرفدار و کاربردی را در حین این پروژه فرا گیرید.

دقت کنید که نمره امتیازی محدود به این لیست نیست و بسته به نحوه پیاده سازی پروژه و خلاقیت های کد و الگوریتمتان، می توانید امتیاز بگیرید.

۲. بارم بندی :

پروژه در مجموع ۱۰۰ امتیاز دارد و تا سقف ۱۵۰ امتیاز می توانید از آن نمره کسب کنید (با توجه به موارد طرح شده در بخش نمره امتیازی مانند کد تمیز، ماژول بندی و ...)

امتیاز	فاز
۱۰	فاز اول
۲۰	فاز دوم
۶۰	فاز سوم
۱۰	فاز چهارم
حداکثر ۵۰	نمره اضافی

۳. توضیحات تکمیلی

- برنامه باید به زبان C نوشته شود. و می توانید از هر یک از نسخه‌ها، شامل آخرین نسخه ی ارائه شده یعنی C18 استفاده کنید.
- در لینک های زیر توضیحات لازم را حتماً مطالعه کنید:
 - <https://www.csie.ntu.edu.tw/~r92092/ref/midi>
 - <http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html>
 - https://en.m.wikipedia.org/wiki/Variable-length_quantity
- دقت کنید که در زمان تحویل پروژه، پروژه شما روی یک سیستم ثالث با سیستم عامل لینوکس کامپایل و اجرا می‌شود پس از همان ابتدا پروژه را روی سیستم عامل لینوکس پیاده‌سازی کنید. همچنین از آدرس‌دهی‌های مطلق در پروژه خود استفاده نکنید.
- استفاده از روش های جایگزین (مانند Memory Allocation) برای پیاده‌سازی هر فاز بلامانع است.
- برون سپاری روشی است که در آن پروژه تعریف شده و سپس برای پیاده‌سازی به افراد دیگری سپرده می‌شود. این روش در صنعت و شرکت های کوچک و بزرگ، موارد استفاده ی خاص خود را دارد و اگر به درستی استفاده شود، نتایج مطلوبی را به همراه خواهد داشت. اما با در نظر گرفتن این که پروژه ی فعلی مربوط به یکی از دروس تخصصی دانشگاه است دانشجویان باید پروژه را شخصاً انجام دهند و هنگام تحویل پروژه بر روی تمام قسمت های کد خود تسلط کامل داشته باشند در غیر اینصورت نمره پروژه را کسب نخواهند کرد.

با آرزوی موفقیت،

تیم حل تمرین مبانی کامپیوتر و برنامه سازی

پاییز ۱۳۹۸