



First Project WorkReport

MICROPROCESSOR AND ASSEMBLY LANGUAGE

DR. SHAKOOR VAKILIAN

FALL - 2025

Nima Makhmali

40212358035 | NIMAMAKHMALI2004@GMAIL.COM

هدف پروژه

هدف از این پروژه، طراحی و پیاده‌سازی برنامه‌ای است که بتواند اعداد را بین مبناهای مختلف (باینری، اوکتال، دسیمال و هگزادسیمال) به صورت دستی و بدون استفاده از توابع آماده‌ی زبان برنامه‌نویسی تبدیل کند. این پروژه برای درک بهتر ساختار سیستم‌های عددی، عملیات بیتی، و مفاهیم پایه‌ای مورد استفاده در زبان اسمبلی و ریزپردازنده‌ها طراحی شده است.

توضیح پروژه

در سیستم‌های دیجیتال، داده‌ها معمولاً در مبنای ۲ (باینری) نمایش داده می‌شوند، در حالی که انسان‌ها برای سادگی از مبنای ۱۰ استفاده می‌کنند. برای کار با حافظه، رجیسترها، و دستورالعمل‌ها، لازم است بتوانیم اعداد را بین مبناهای مختلف تبدیل کنیم.

مبنا	نام	محدوده ی ارقام	مثال
2	باینری	0, 1	1011_2
8	اوکتال	7-0	73_8
10	ده دهی	9-0	123_{10}
16	هگزادسیمال	0-9 , A-F	$1A3_{16}$

روش کلی تبدیل:

1. تبدیل عدد از مبنای مبدأ به ده‌دهی: (Base \rightarrow 10)

با استفاده از رابطه:

$$value = \sum_{i=0}^{n-1} digit_i * base^{(n-i-1)}$$

2. تبدیل عدد ده‌دهی به مبنای مقصد: (10 \rightarrow Base)

با تقسیم پی‌درپی عدد بر مبنا و ذخیره‌ی باقیمانده‌ها از انتها به ابتدا.

3. تبدیل ده‌دهی به مبنای دیگر

در تبدیل یک عدد از مبنای ده (ده‌دهی) به مبنای دلخواه (۲، ۸ یا ۱۶)، از روش تقسیم و باقیمانده‌ی پی‌درپی برای بخش صحیح و از روش ضرب متوالی در مبنا برای بخش اعشاری استفاده می‌شود.

بخش صحیح:

1. عدد صحیح را بر مبنای مقصد تقسیم می‌کنیم.
2. باقیمانده‌ی هر مرحله، رقم متناظر در مبنای جدید است.
3. باقیمانده‌ها را از انتها به ابتدا می‌نویسیم تا عدد در مبنای جدید به‌دست آید.

بخش اعشاری:

1. بخش اعشاری عدد را در مبنای مقصد ضرب می‌کنیم.
2. بخش صحیح حاصل ضرب را به‌عنوان اولین رقم بعد از ممیز در مبنای جدید می‌نویسیم.
3. بخش اعشاری باقی‌مانده را مجدداً در مبنا ضرب می‌کنیم.
4. این فرآیند تا زمانی ادامه می‌یابد که عدد اعشاری به صفر برسد یا به تعداد رقم دلخواه اعشار برسیم.

اعداد منفی:

در صورتی که عدد ورودی دارای علامت منفی (-) باشد، برنامه ابتدا مقدار مطلق عدد را به مبنای جدید تبدیل کرده و در پایان، علامت منفی را به ابتدای نتیجه اضافه می‌کند.

4. پشتیبانی از اعداد منفی

اعداد منفی در رشته ورودی با علامت (-) در ابتدای عدد مشخص می‌شود. برنامه با شناسایی علامت ابتدا مقدار مطلق را تبدیل کرده و سپس علامت منفی را به نتیجه اضافه می‌کند.

شرح الگوریتم و توابع

نام تابع	توضیح عملکرد
<code>char_to_value_for_base(ch, base)</code>	مقدار عددی یک رقم در مبناهای مختلف
<code>value_to_char(val)</code>	برگرداندن کاراکتر متناظر با مقدار عددی
<code>to_decimal(number, base)</code>	تبدیل رشته عددی از مبنای مشخص به مبنای ۱۰
<code>from_decimal(decimal_number, base_to)</code>	تبدیل عدد ده‌دهی به مبنای دیگر
<code>convert(number, base, to_base)</code>	تابع اصلی تبدیل بین دو مبنا
<code>parse_fractional_part()</code>	تبدیل بخش اعشاری (اضافه شده در نسخه جدید)
<code>main()</code>	کنترل تعامل با کاربر و نمایش نتایج

نمونه اجرای برنامه

مثال ۱:

```
Input Number: 1101.11
Base from: 2
Base to: 10
Result:  $(1101.11)_2 = (13.75)_{10}$ 
```

مثال ۲:

```
Input Number: -255.5
Base from: 10
Base to: 16
Result:  $(-255.5)_{10} = (-FF.8)_{16}$ 
```

مثال ۳:

```
Input Number: 2A
Base from: 16
Base to: 10
Result: (2A)16 = (42)10
```

تحلیل الگوریتمی

مرحله	زمان اجرا	توضیح
تبدیل به مبنای ده	$O(n)$	تعداد ارقام عدد ورودی
تبدیل از مبنای ده	$O(\log_{\text{base}} N)$	وابسته به اندازه عدد
پشتیبانی از اعشار	$O(k)$	تعداد رقم‌های اعشاری که محاسبه می‌شوند.
مجموع پیچیدگی	$O(n + \log_{\text{base}} N + k)$	برای اکثر اعداد کارایی بالا دارد.

نتیجه‌گیری

این پروژه نشان داد که با طراحی دقیق توابع، می‌توان تبدیل بین مبناها را به صورت عمومی و ساخت‌یافته پیاده‌سازی کرد. افزودن پشتیبانی از اعداد اعشاری و منفی، برنامه را به سطح بالاتری از کاربرد رسانده و آن را به یک ابزار آموزشی مفید برای درس‌های نظریه زبان‌ها، سیستم‌های عددی و طراحی الگوریتم تبدیل کرده است.