



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 مبنای مختلف (باينری، اوكتال، دسيمال و هگزادسيمال) به صورت دستی و بدون استفاده از توابع آماده‌ی زبان برنامه‌نويسی تبدیل کند. این پروژه برای درک بهتر ساختار سیستم‌های عددی، عملیات بیتی، و مفاهیم پایه‌ای مورد استفاده در زبان اسambilی و ریزپردازندۀ طراحی شده است.

توضیح پروژه

در سیستم‌های دیجیتال، داده‌ها معمولاً در مبنای ۲ (باينری) 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 مبنای مختلف تبدیل کنیم.

مثال	محدوده ارقام	نام	مبنای
1011 ₂	0, 1	باينری	2
73 ₈	7-0	اوكتال	8
123 ₁₀	9-0	دهی	10
1A3 ₁₆	0-9 , A-F	هگزادسيمال	16

روش کلی تبدیل:

1. تبدیل عدد از مبنای مبدأ به دهدھی: (**Base → 10**)

با استفاده از رابطه:

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

2. تبدیل عدد دهدھی به مبنای مقصد: (**10 → Base**)
با تقسیم بی‌درپی عدد بر مبنای ذخیره‌ی باقیمانده‌ها از انتهای ابتداء.

۳. تبدیل ددهی به مبنای دیگر

در تبدیل یک عدد از مبنای ده (دهدهی) به مبنای دلخواه (۲، ۸ یا ۱۶)، از روش تقسیم و باقیماندهی پی درپی 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 انتهای ابتدا می‌نویسیم تا عدد در مبنای جدید به دست آید.

بخش اعشاری:

1. بخش اعشاری عدد را در مبنای مقصد ضرب می‌کنیم.
2. بخش صحیح حاصل ضرب را به عنوان 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 بررسیم.

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.

۴. پشتیبانی از اعداد منفی

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

شرح الگوریتم و توابع

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

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

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

نتیجه‌گیری

این پژوهه نشان داد که با طراحی دقیق توابع، می‌توان تبدیل بین مبنایها را به صورت عمومی و ساخت‌یافته پیاده‌سازی کرد. افروزنده 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.