

مهر ای پیاده سازی زبان های برنامه سازی

21 Apr. Sun. 2019
۱۵ شعبان ۱۴۴۰
اردیبهشت
۹۸/۲/۰۱

۱- رونق مادی زبان برنامه سازی خود ۱۰٪ رونق مادی زبان برنامه سازی

۲- رونق مادی زبان برنامه سازی خود ۱۰٪ رونق مادی زبان برنامه سازی

۱- رونق مادی زبان برنامه سازی خود ۱۰٪ رونق مادی زبان برنامه سازی
(write ability) مطرح خود منظور از قابلیت خواندن این است که به صورت فله ای دستورات زبان خوانا باشد
و نامبر زبان مابین دستورات آن مقصود معنای مفهوم آن باشد و نیاززی به رجوع مقدمات زبان نداشته
باشد که منظور از مقدمات همانند یک کتاب آموزشی یا هر زبان باشد به این وسیله می توانیم خود
استفاده از **self document** کنیم. منظور از قابلیت نوشتن در زبان برنامه سازی
می تواند به این صورت مطرح شود که پیاده سازی دستورات زبان برنامه نویسی آن به راحتی صورت گیرد
و به این راحتی می تواند عوامل بودن تعداد ناایتم های دستورات زبان در نظر گرفته شود و معمولاً ویژگی های
خواندن و نوشتن در زبان های برنامه سازی یک جا با هم فراهم می شود بدین معنا که هر چه قابلیت
خواندن بیشتر باشد قابلیت نوشتن آن کم تر می باشد.

مثال: فرض کنید می خواهیم یک متغیر α از نوع عددی صحیح در زبان های ++C و پایتون تعریف کنیم که صورت

فله ای این دستورات در زبان قدرتمند به صورت زیر باشد

```
C++: int alpha;
```

Pascal: alpha: integer;

با توجه به دستورات فوق می توان گفت که در زبان ++C نسبت به زبان پایتون دارای قابلیت نوشتن ساده

تر در زبان پایتون نسبت به ++C دارای قابلیت خواندن است که با توجه به این مثال می توانیم مابین قابلیت

خواندن و نوشتن آفات می توانیم

۲- قابلیت خواندن orthogonality: منظور از قابلیت خواندن در زبان های برنامه سازی بهضمان

ویرایش زبان برنامه سازی ضد این است که زبان برنامه سازی



ولایت حضرت قائم (عج) (۱۳۹۰ هـ.ق) (مطابق) و روز جهانی مستطیل - روز بزرگداشت سعدی

مست از می و میخواران از نوگس همیش مست

در دیر معیان آمد یاربر قدحی در دست

۲- قابلیت خواندن

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

April

این قابلیت را برای برنامه نویسی زبان فراموش نکنیم بعد از آن در هر دو زبان

22 Apr, Mon, 2019

۱۶ شهریور ۱۴۴۰

۲

دوشنبه اردیبهشت

۹۸/۰۲/۰۲ هفته ۳۲۲-۳۲۳

منطق زبان را با یادگیری ترکیب کند به گونه ای که ترکیب واحد هم برای

ی بیاییم زبان و هم برای برنامه نویسی آن دارای معنا و مفهومی باشد.

سوال ۸ دستگیر $(a+b) \text{ write } (a \times b) \text{ if}$ در زبان با مثال را در نظر بگیریم به گونه ای که در زبان
دستگیر a و عملگر رابطه ای مخالف بودن دو مقدار دستگیر a و b write و عملگر حسابی جمع با هم
ترکیب شدند.

۳- بتوانیم در زبان برنامه نویسی با تعداد مقولود کمتری فوایدی قدر با یاد سازی خاشیم که این امر
محو لا در زبان های سطح بالا فراموش می شود.

سوال ۹ فرض کنید می خواهیم در زبان ما سین و اسمبلر جزای بنویسیم که ما خاشیم
بیجا گاهی مناسب در عدد را در عددی دریافت کند و حاصل جمع آن ها را با خاشیم بیجا مناسب
دلیله در خروجی نمایش دهد که ترکیب به محافظه گد نیاز خواهیم داشت ولی همین برنامه را فرست
کنیم می خواهیم در زبان سطح بالا ما $C++$ یا C قرار بنویسیم

که تعداد گد های آن نسبت به زبان سطح پایین اسمبلر قابل به $\frac{1}{10}$ می باشد. تغییر بسیار کند.
۴- و منبع فراموش مناسب را در مثال به فعال دستگیر a در زبان برنامه نویسی فعلی
نظم گرفت.

سوال ۱۰ محیط زبان های Borland, Pascal را در نظر بگیریم به گونه ای که به صورت یک صفی آبی رنگ
مرا با یک مثال گای نزدیک به تعداد قابل محیط فوق زبان و مثال بیست را در نظر خواهیم
گرفت که دارای وجود کمترین مقولود تری می باشد که این امر محولا در زبان های سطح بالا
فراموش شده است.

QR code and footer text: به جان خواجه و حق قدیم و عهد دوست

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰	۳۱
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

۱- قابلیت عمل یا Portable بودن معنای یک زبان

برنامه سازی نیابتی وابسته به ماشین یا کامپیوتری خاص یا فقط در این معنای امر برنامه های عمل سیستم کامپیوتری در یک زبان برنامه سازی نوشته می شود و انتقال به کامپیوتری دیگر انتقال دهیم که این امر برای برنامه نویسان زبان مطلوب نمی باشد.

۲- با این بودن هزینه زبان برنامه سازی؟

ظلم رقم

درمان زبان های برنامه سازی هزینه های قابل از ۳۰ تا ۳۰۰ هزار دلار مورد بررسی قرار داده

۱- هزینه ترجمه

۲- هزینه اجرا

۳- هزینه نگهداری

هزینه ترجمه معمولاً در زبان های سطح می شود که در محیط های کاربردی مورد استفاده قرار می گیرد زیرا برنامه های نوشته شده در این دسته از زبان ها معمولاً دارای فضای باقی مانده برای توسعه بیشتر و امکان با تغییر هم خواهند شد و هزینه اجرای زبان های مهم فضا محدود در طول روز به دفعات زیاد اجرا می شود **نظریه نگهداری** همان طور که می دانید در طول عمر یک محصول نرم افزاری هزینه نگهداری می تواند دارای

اهمیت بیش از حدی باشد زیرا دوره نگهداری محصول نرم افزاری در طول عمر آن بیشترین دوره را به خود اختصاص می دهد بنابراین محصول خوب خواهد بود که دوره نگهداری آن دارای کمترین هزینه باشد و نگهداری یک محصول نرم افزاری می تواند به ۳ نوع نگهداری نگهداری (perfective)

نگهداری اصلاح فرین (corrective) نگهداری تطبیقی (Adaptive) بخش خایه پس از گذشت زمان کرده می تولید گفته می - محصول نرم افزاری بتواند قابلیت های جدیدی را به یک محصول نرم افزاری اعمال کند تا در اعمال یک با احتمال داده خود در یک زبان برنامه سازی که امکان

نگهداری تولید گفته می شود

M	T	W	T	F	S	S
	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					



اصلاحی: همان طوره که می دانید به محصل نهم اموزی و قبل از
و آنگاه دارد فظاهای

24 Apr. Wed. 2019

۴

۱۸ شعبان ۱۴۴۰

اردیبهشت

۹۸/۲/۰۴ هفته ۲۳۰-۲۳۱

تعمیر به محصل نهم

از محصل در آن مرحله به حال ماند و با مرور زمان و کار کردن کار به آن با محصل فظاها نمایان
می شود بنا به این وظیفه مقرره توسعه دهنده می محصل می باشد که فظاهای موجود را اصلاح نماید
تطبیقی: فرض کنید زبان برنامه سازی طراحی و تولید شود و با ارائه یک سیستم عامل
جدید این زبان قابلیت اجرا شدن در آن سیستم عامل را نداشته باشد بنا به این
وظیفه توسعه دهنده آن زبان برنامه سازی می باشد که تغییراتی در آن اعمال نماید که زبان های
برنامه سازی قابلیت تطبیق با محصل جدید را داشته باشند.

دلایل صلاحیت زبان های برنامه سازی:

- انتخاب بهترین زبان: اجرای نوشتن یک برنامه به عنوان مثال فرض کنید بنویسیم
یک وب سایت طراحی کنیم و با توجه به شناخت زبان های برنامه سازی که با مطالعه آن
عامل می شود بهترین زبان را برای طراحی آن وب سایت انتخاب کنیم.
- انتخاب بهترین ساختار داده: برای نوشتن برنامه در یک زبان خاص به عنوان مثال
فرض کنید در زبان ++C بنویسیم که یک عدد حقیقی را از ورودی دریافت
کنند و معادل دهمی آن در خروجی نمایش دهند. راهکار بیس یا استفاده برای نوشتن این برنامه
اعمال ۱۰۰۰۰ دستور if خواهد بود که آلفا چشم زبان برنامه سازی گفته است و الله اعلم
برای نوشتن این برنامه استفاده از مفهوم تابع خواهد بود که در این راهکار برنامه خروجی را می توان
با ۷۰ الی ۷۰۰ خطا کم باره سازی نمود.

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰
۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰



مدل‌های زبان های برنامه سازی

۱- مدل دستوری: این مدل زبان های برنامه سازی ترکیبی

از دستورات مختلف می باشد و این دستورات به ترتیب از دستوره اول تا دستوره n اجرا می شوند
 ۲- مدل فلهی: در این مدل از زبان های برنامه سازی به صورت زیر می توانیم گفتند

```

instruction 1;
instruction 2;
...
instruction n;
    
```

۲- مدل تابعی: در این مدل از زبان های برنامه سازی درون یک برنامه می توانیم توابع

مختلف تعریف کنیم و نوشتن یک برنامه با مفهوم تابع می تواند برای
 فرآیندی مختلفی باشد مانند C++ و پاسکال Pascal به صورت فلهی برنامه درون
 این مدل از زبان های برنامه سازی می توانیم شکل زیر مطرح نمود

۶
 26 Apr. Fri. 2019
 ۲۰ شعبان ۱۴۴۰
 اردیبهشت چهارشنبه
 هفته ۲۳۷ | ۵ - ۲۳۸ هـ ۹۸/۰۲/۰۶

function n (..... (function 2 (function 1 (Data)))

و مشابه با دستوره فوق می توانیم فرقی کنیم که مدل یک برنامه

n عدد تابع برنامه می باشد function 1 تا function n درون

یک برنامه می باشد و n عدد تابع فوق به صورت متداخل از function n تا function 1

درون یک برنامه می باشد و فرآیندی می شود که در بالا ترین سطح function 1 یا Data

فرآیندی خواهد بود و فرقی این تابع به دنبال بار اجرای function 2 می باشد

می شود و این روند تا function n تکرار می شود



M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

۳- مدل هوش مصنوعی: در این مدل از نتایج برنامه سازی

27 Apr. Sat. 2019
۲۱ شعبان ۱۴۴۰
شبه اردیبهشت
۹۸/۰۲/۰۷ هفته ۲۲۷-۵۲۸

۸۸ یک برنامه می تواند تعالی منقش از دستورات شرطی در نظر گرفته شود
۸۹ ما در زبان C و در زبان های matlab و Python نیز قواعد منقش است. یادگیری

۹۰ اگر در سیستم های هوش مصنوعی برای برنامه نویسی فراموش شده است. می توانیم صورت
۹۱ تلاشی یک برنامه در مدل هوش مصنوعی را به شکل زیر در نظر بگیریم.

if conditional then action 1

11 if conditional then action 2

۱
۱۲
۱۳

if conditional n then action n

۱۴ ۴- مدل شیء تراشه در این مدل زبان های برنامه سازی می توانیم اشیاء و منقش تعریف کنیم
۱۵ و سپس قواعد است که در این اشیاء هوش مییم ما در زبان های Python و C# می توانیم
تجزیه عمل اول

۱۶ مثال: دستوراتی در زبان ++C بنویسید که همین آن ها به یک واحد متغیر x اضافه کند.

- ۱۷ (۱) $x = x + 1$
- (۲) $x ++$
- (۳) $++x$
- ۱۸ (۴) $x + = 1$

۱۹ سوال: در یک زبان تابعی می توانیم یک واحد متغیر را در یک تابع تعریف کنیم و آن را به یک واحد متغیر اضافه کنیم.
۲۰ و یک تابعی بنویسید.

→	←	→	←	→	←	→	←
۶	۵	۴	۳	۲	۱	۷	۸
۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶
۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳
۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰



دو نیمی جدول نقل

خدا چو صورت اسری دلگشا...

دسته چهارم

نقد یا Intax 5.5 : صورت فلهری دستورات که بر اساس آن خود

گردیده برای عمل با عمل در زبان های مختلف ممکن است خود متفاوت داشته باشد

نقد عمل مثال برین فایده خواهد بود که برای و اعمدهری به نام α از نوع عددی صحیح در جدول

برنامه سازی C++ و Pascal توهم فایده که دارای خود متفاوتی به صورت زیر می باشد:

```
C++: int a[10];
```

```
Pascal: a=array [1..10] of integer;
```

مانند به دستورات حقوق می توانیم بگوییم زبان پاسکال نسبت به C++ دارای قابلیت

خواندن در C++ نسبت به پاسکال دارای قابلیت نوشتن می باشد.

برای مصفوم خود می توانیم معیارهایی به صورت زیر مطرح کرد ۱۸- قابلیت خواندن و نوشتن

۲- عدم وجود انبعاث ۳- بدین معنای صورت فلهری دستورات برنامه با سبکی به گونه ای باشد

که از دید برنامه نویسی و کامپایلر زبان با معنی ایها و ابجکت شود به عنوان مثال Fortran

دارای ابجکت می باشد که می از ابجکت آن به پاسکال بدین خصوصیت فواید تابع و اشاره

به فایده ای از آن که چون آن می باشد که در زبان پس از آن (Pascal) ابجکت مورد نظر

به طریقی جدید بدین صورت که برای اشاره به فایده ای از آن که در [] استفاده شده است

۳- سهولت بازیابی ۸- صورت فلهری دستورات که زبان می تواند سهولت بازیابی را

برای برنامه نویسان فراهم نماید و این سهولت زمانی فراهم می شود که زبان دارای قابلیت

خواندن باشد ۴- سهولت ترجمه ۸- در زبان های برنامه سازی هر چه قابلیت نوشتن بیشتر باشد

سهولت ترجمه برای فایده های تطبیق لغوی یا (Lexical Analyser) تطبیق لغوی کامپایلر فراهم

می نماید زیرا تطبیق لغوی کامپایلر زبان کمتری را می توانم



	M	T	W	T	F	S	S
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30						

CMP = compiler

JZ = jump zero

JNZ = jump not zero

ZF = zero flag

10

30 Apr, Tue, 2019

۲۴ شعبان ۱۴۴۰

ارديبهشت سه شنبه

هفته ۱۷ | ۳۴۴ - ۹۸/۰۲/۸۰

اعمال منطقی و اعمال محاسبه منتهی از خالص داده است

به عنوان مثال فرض کنید خواهیم در زبان ماشین و اسمبلر برنامه بنویسیم

متغیرات در زبان Ax و Bx را بر روی حافظه محاسبه می‌کنیم و آنرا این عمل از دستور $CMP\ Ax\ Bx$

استفاده می‌کنیم که اگر دستور این دو متغیرات نام برابر باشد حاصل عمل تفریق فوق صفر

خواهد شد و با دستور شدن این عمل صافی $ZF=1$ می‌شود و در دستورات JZ و JNZ می‌توانیم

فرآیندی

عملت اول برنامه استفاده می‌کنیم

۱۳ کنترل ترتیب: زبان های برنامه سازی دارای محاسبه کنترل ترتیب دستورات

برنامه‌ها با توجه به این نوع از دستورات فرآیند دارای انواع مختلفی خواهد بود به عنوان مثال

زبان C++ را در نظر بگیرید که در آن این زبان خواهد بود برای بنویسیم که عددی را

از ورودی دریافت کرده و قدر مطلق آن در خروجی نمایش دهد

`int x;`

`cin >> x;`

`if (x <= 0) cout << x;`

`else`

`cout << (-x);`

۱۵ بازگردد به برنامه فوق به کمک دستور `if` همراه `else` عمل کنترل ترتیب صورت

۱۶ گرفته است و خروجی داریم از دستور کنترل ترتیب دستور سوئیچ در این زبان می‌توانیم

که به کمک این دستور می‌توانیم از `if` های متناظر و توابعی بهره‌مند شویم

۱۷ ملاحظه کنید دستبندی بردار: تمامی اشیاء دایره مطرح شده درون زبان های برنامه سازی

درون حافظه جای می‌گیرند که ملاحظه کنید دستبندی بردار درون حافظه که در آن هست به عددی

در فضای ۱۶ خواهد بود و باید بدانیم که x شروع می‌شود به عنوان مثال برنامه نویسی را در نظر بگیرید

که در این سری داده x درون حافظه قرار می‌گیرد



روز ملی خیریت

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

مرا فشان دل از ره تو را چه افتادست

برو به کار خود ای واعظ این چه فریادست

```

#include <iostream>
using namespace std;
int main() {
    int x, *p;
    08 cout << endl; << "Enter number = ";
    cin >> x;
    p = &x;
    09 cout << endl; << "
    cout << "it" << "x = " << x;
    cout << endl; << "it" << "the address of x is = " << p;
    10 return 0;
}

```

11
 12
 13

۵- **مفهوم عملیاتی** : مفهومی که در اصل شامل تمام متغیرهای برنامه است که می‌توانیم به سبب دسترسی به آن‌ها از آن‌ها برای برنامه‌نویس استفاده کنیم. این متغیرها می‌توانند با زبان‌های خارجی خود ارتباط برقرار کنند مانند متغیرهای **input** و **output**.

۶- **مدیریت حافظه** : حافظه‌ای که در سیستم اصلی درون سیستم‌های مدیریت مطرح می‌باشد. در این است که مولفه‌های CPU تا حد امکان منقول باشند و به یکدیگر واگذار شوند. در هر قسمتی از این اصل مطرح می‌شود که در آن سرعت پردازش CPU در مقایسه با دستمای برنامه درون حافظه اصلی می‌باشد که تحت این تناقض قرار می‌گیرد. مطرح می‌شود راهکار اول (معمولاً `malloc` یا `calloc`) می‌باشد که معنای واقعی CPU را واسطه اجرای کدهای (Process) برنامه‌های درون حافظه

→	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳
→	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶
→	۲۷	۲۸	۲۹	۳۰	۳۱								

نیاز داشته باشد زیرا در سیستم‌های توزیع‌شده و غیره نیز قابل استفاده است.



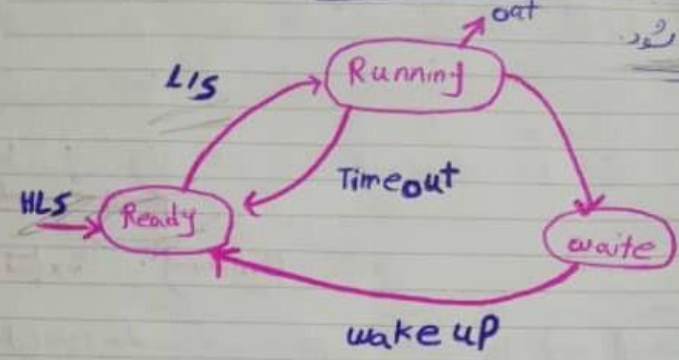
تا بر زلف تو در دست نسیم افتادست

دل سودازده از غصه دو نیم افتادست

دوره - wait منتقل می کند و فرکانس جدیدی را اجرا می نماید

تاریخاتی که برای فراگرفته شدن wake up تلفظ داده می شود

در درون حافظه قابل شود



۱۲ حافظه را که برای نوع این مشکل از حافظه کش یا Cache memory خواندن CPU اتفاق می افتد

۱۳ به این صورت که اگر CPU به داده ای نیاز داشته باشد ابتدا داده مورد نظر را از درون

حافظه کش خوانده می شود در صورت معصوم بودن داده یک بیت امضا یا پرچم داده مورد نظر

۱۴ مورد را قاره فراهم می کند و در صورت معصوم شدن داده درون حافظه کش miss رخ می دهد

داده حقوق به همراه داده های صحیح از درون حافظه کش می خواند

مشکل می شود زیرا ممکن است فرکانس جدیدی را اجرا نماید

داده های مغفول CPU نیاز به انتقال به پرچم داده های صحیح

CPU درون حافظه کش می تواند از طریق رابطه زیر

محاسبه شود

$$hit\ Ratio = \frac{hit}{hit + miss}$$

نکته: نرخ انتقال اطلاعات با CPU و حافظه کش word یا کلمه



بازار و با این حافظه کش و حافظه کشی است و در حافظه کش

M	T	W	T	F	S	S
	1	2	3	4	5	6
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

برنامه ها تا قطعی مقدار یافته اند مقدار می گیرند

مثال در $x=10$

به عنوان مثال برنامه ای ساده برآورد مقدار یک عدد در مقدار از ورودی دریافت خود و در یک

مغز و تابع حاصل می آید اما در برنامه در فرآیند خروجی به چاپ می آید.

```
#include <iostream>
```

```
float sum (float, float); // Declaration
```

```
int main () {
```

```
float x, y;
```

```
cin >> x >> y; Actual parameter
```

```
cout << sum (x, y); // call
```

```
return (0); }
```

```
float sum (float a, float b) { // Define
```

```
return (a+b);
```

format parameter

حال چگونه در برنامه فوق مشخص شده است پارامترهای که در فراخوانی تابع ظاهر می شوند پارامترهای

واقعی و پارامترهای که در تعریف تابع نمایان می شود پارامترهای معاری خواهند بود (a, b)

و اولین اتفاق هر دو فراخوانی یک بر سر برنامه سطح می شود تا طر با این این خود دست پارامتر

خواهد بود که تا طر فوق در زبان های برنامه معاری به در صورت تا طر معین و تا طر با ناگه باشند

که در حالت تا طر معین اولین پارامتر واقعی با اولین پارامتر معاری



روز بزرگداشت شیخ صدوق

دومین پارامتر واقعی با دومین پارامتر معاری و ...

شعباد خانه پرور ما از که کمتر است

باغ موا چه حاجت سرو و صنوبر است

M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

بارانتر را قس با n این بارانتر سازی متناظر باشد که در مثال فوق

بارانتر a بارانتر b متناظر خواهد شد که این

زمان های برابر سازی از متناظر موقعیتی اتفاق دهی نماید و متناظر یا ناگ بارانترهای واقع

و معاری بر اساس نامشان متناظر خواهد شد.

مثال: انقضاهای مطرح به همراه زمان $K8 = K + 2$ و مقومشان را برای دستنویز $K8 = K + 2$

۱- نوع تغییر در زمان تر حجم زمان

۲- مقدار تغییر در زمان اجرا

۳- مثال تلاصق عملیه حسابی جمع در زمان $K8 = K + 2$

۴- عملیه در عملیه جمع در زمان تر حجم زمان $K8 = K + 2$ این عملیه در زمان $K8 = K + 2$ با سلال یک عملیه صد

رینتی محسوب می شود و به حسب این که محلودهاش از چه نوع باشد امکان مختلف

را انتخاب نمود در زمان فوق آنرا محلودهای عملیه جمع عددی صمیم باشند عمل جمع صمیم

آنرا از نوع اعشاری باشند عمل جمع اعشاری که آنرا از نوع متعدهای باشند عمل اجتماع دو مجموعه

و آنرا از نوع رشته ای باشند عمل الفارق صورتی و اینها هم در این بنا بر این $K8 = K + 2$

فوق باستی در زمان تر حجم زمان نوع محلودهای این عملیه مشخص شود تا تا بنا بر این

زمان بتواند محلود مناسب را داشته باشد

۵- مثال تلاصق عدد ۲۰ در تغییر زمان و جزئیات مربوط به آن مانند تقاربت های

افصاضی در زمان پیاده سازی زمان مشخص خواهد شد

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰
۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰



۶- محدوده مقادیر معادله برای تغییر k در زمان t یا در زمان t یا در زمان t

$f \text{ bat } a ;$

تست: دستور بدست آوردن زمان t در نظر بگیرید

$a = a + at$

انضاد دفع مقید و مجموع مقادیر ممکن مقید a برتر است در هر طریقی ممکن می‌شود

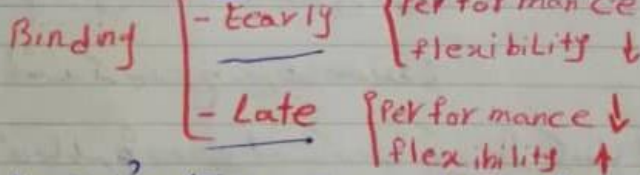
الف) ترجمه زمان - توقف زمان

ب) ترجمه زمان - یا در زمان سازی زمان

ج) ترجمه زمان - ترجمه زمان

د) یا در زمان سازی زمان - توقف زمان

انواع انضاد: انضاد می‌تواند به دو صورت زودرس و دیررس مطرح شود



انضاد زودرس در زمان ترجمه مطرح می‌شود ولی انضاد دیررس در زمان اجرا مشخص خواهد شد

در انضاد زودرس با توجه به اینکه در زمان کامپایل ترجمه سرعت می‌گیرد دارای کارایی بالا می‌باشد

انحطاف پذیری پایین خواهد بود در انضاد دیررس با توجه به اینکه در زمان اجرا انجام می‌شود

دارای کارایی پایین ولی انحطاف پذیری بالا خواهد بود



M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

(م.ع.ه)

8 May Wed 2019

۱۸

۳ رمضان ۱۴۴۰

ارديبهشت

چهارشنبه

۹۸/۲/۸۸ هفته ۵۴۹۱۸-۲۱۶ هـ

۰۸ **الف** در راهی انقضاء زبان ها کدام فرقیه صحیح است؟
 ۰۹ **الف** در زبان های با مقید سازی دیر هنگام قابلیت اغلاف پذیری در سرعت اجرای برنامه مناسب تر است

۱۰ **ب** در زبان های با مقید سازی دیر هنگام قابلیت اغلاف پذیری بیشتر در مقید سازی زود هنگام سرعت اجرای برنامه ها مناسب تر است.

۱۲ **ج** در زبان های با مقید سازی دیر هنگام قابلیت اغلاف پذیری کم تر در مقید سازی زود هنگام سرعت اجرای برنامه ها مناسب تر است.

۱۳ **د** در زبان های با مقید سازی زود هنگام قابلیت اغلاف پذیری در سرعت اجرای برنامه ها مناسب تر است.

طرحه هفتم

۱۴ **انواع مفعولها:** برای اینکه برنامه نوشته شده در یک زبان بر نامه سازی به زبان قابل معم برای سیستم کامپیوتری ترجمه شود مفعولهای مختلفی است انجام این عمل در کامپیوتر می باشد که به شرح زیر خواهد بود:

۱۶ **۱- کامپایلر:** مفعولی است که معمولاً یک زبان سطح بالا را به زبانی نزدیک به زبان ماشین واسطی تبدیل می نماید تا بر این می توانیم به مفعول کامپایلر به نحوی عمل می توانند که برنامه سطح بالا را به زبان ماشین واسطی (اوه) تبدیل نماید.

۱۸ **۲- Assembler:** مفعولی است که در زبان صاف آن زبان ماشین واسطی و زبان مفعول آن زبان ماشین واسطی (اوه) می باشد.

→	۱	۲	۳	۴	۵	۶
→	۷	۸	۹	۱۰	۱۱	۱۲
→	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
→	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
→	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰
→	۳۱	۳۲	۳۳	۳۴	۳۵	۳۶



روز بیماری های خاص و ضعف العالج

حال دل با تو گفتم هوس است

خبر دل شنفتم هوس است

۹۰۵۸

8 May Wed. 2019

۱۸

۳ رمضان ۱۴۴۰

چهارشنبه اردیبهشت

۹۸۰۲۸۸ هفته ۵۴۹۱۸-۲۱۶

۸ **الف** در راهی انقیاد زبان ها نگاه فرمایید جمع است ؟
 ۰۸ **الف** در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری در سرعت اجرای
 ۰۹ برنامه مناسب تر است

۱۰ **ب** در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری بیشتر ولی مقید سازی
 زود هنگام سرعت اجرای برنامه ها مناسب تر است.

۱۲ **ج** در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری کم تر ولی مقید سازی
 زود هنگام سرعت اجرای برنامه ها مناسب تر است.

۱۳ **د** در زبان های با مقید سازی زود هنگام قابلیت انعطاف پذیری در سرعت اجرای
 برنامه ها مناسب تر است.

طرحه هشتم

۱۴ **انواع معضرها** برای اینکه برنامه نوشته شده در یک زبان بر پایه سازی به زبان قابل
 معمم برای سیستم کامپیوتری ترجمه شود معضراتی مثل عبارت انبیا این محل درگیر
 می باشند که به شرح زیر خواهم نمود :

۱۶ **۱- کامپایلر** معضری است که معمولاً یک زبان سطح بالا را به زبانی نزدیک به زبان ماشین
 واصل می تبدیل می نماید تا بر این می توانیم به معمر کامپایلر به نحوی می توان تبدیل برنامه
 سطح بالا را به زبان ماشین واقعی (اده) تبدیل می ی

۱۸ **۲- Assembler** معضری است که زبان صبا آن زبان ماشین واصل
 در زبان معقد آن زبان ماشین واقعی (اده) می باشد.

۱	۲	۳	۴	۵	۶
۷	۸	۹	۱۰	۱۱	۱۲
۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
۲۵	۲۶	۲۷	۲۸	۲۹	۳۰
۳۱	۳۲	۳۳	۳۴	۳۵	۳۶



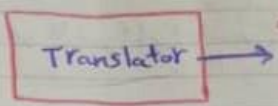
روز بیماری های خاص و ضعف العلام

حال دل با تو گفتم هوس است

خیر دل شغفتم هوس است

Source

Program →



Object (Destination)
Program

۳- پیش پردازنده یا پیش پردازشگر (preprocessor) نوعی مفسر است که زبان مبدأ و مقصد آن یکسان است به عنوان مثال در تعریف include دستهای برای ++C و ++C را نامبرد که مقصد این دستورات زبان ++C باشد.

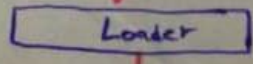
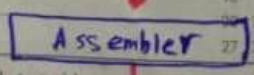
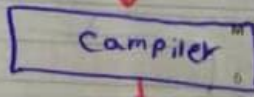
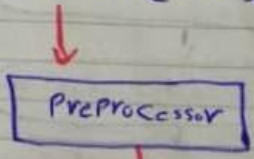
۴- بارنده یا Loader: آدرس تولید شده توسط مفسر را با آدرس CPU برای مفسرهای مختلف برنامه آدرس های معاری می باشد بدین معنای این آدرس ها ممکن است توسط واحد حافظه قابل رویت یا تعداد این رویت های مفسر بارنده این آدرس های معاری تولید شده را به آدرس های واقعی درون واحد حافظه تلاشت نماید.

۵- Linker: وظیفه این مفسر ایجاد ارتباط با لینک مفسرهای مختلف یک برنامه است تا آن برنامه در قالب یک برنامه اجرایی واحد تبدیل فاشم به عنوان مثال ترتیب استفاده از

مفسر های مطرح شده عمدتاً اجرا می کنند یک برنامه در زبان ++C به صورت زیر می باشد

Source

Program (C++)



M	T	W	T	F	S	S
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

وقت گل خوش باد کز وی وقت میخواران خوش است

صحن بستان ذوق بخش و صحبت یاران خوش است



نارطای ما میایم که هر دو میایم نارطی شش نارطی باقی

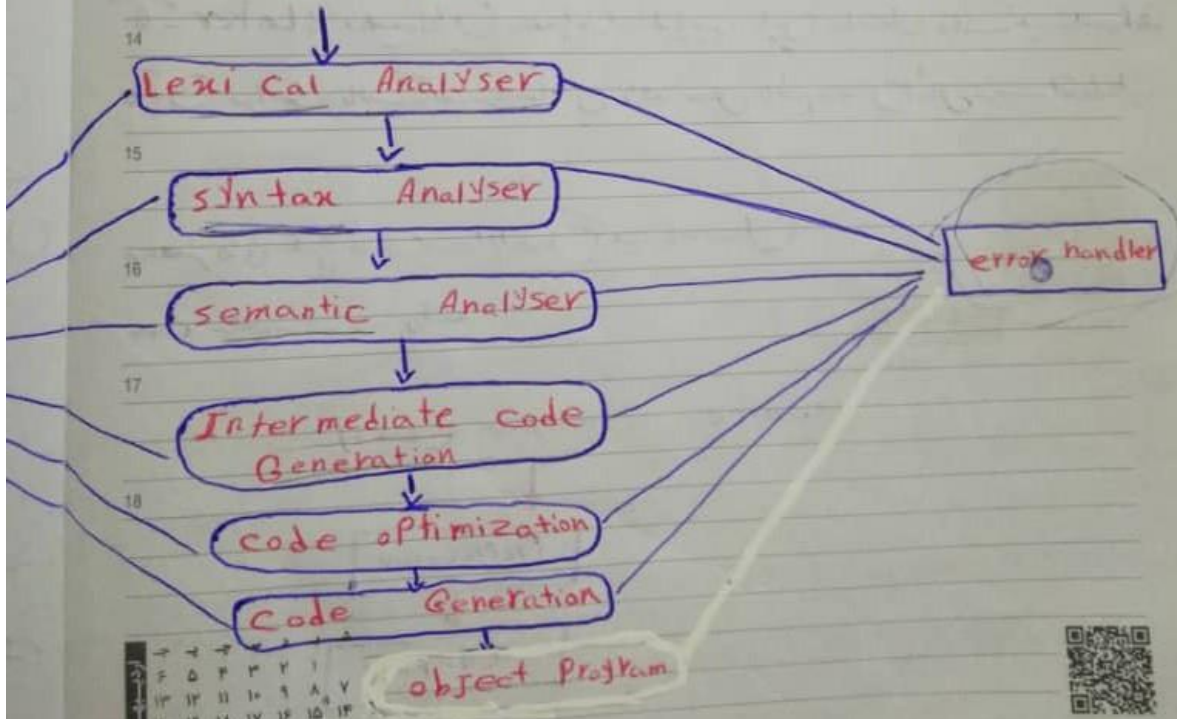
۹. **ارعدادندار تطبیق معنوی (Lexical Analyser)** : تطبیق معنوی (Syntax Analyser)

۱۰. **تطبیق معنایی (Semantic Analyser)** : تولید کد میانی یا مابین

۱۱. **بهرینه سازی Code** و تولید کدهای میانی یا Code Generation و منحصراً ما میایم ملازمه شش نارطی

۱۲. **نامبرده شده نارطی در مؤلفه دیکشنری** یا آنچه که عبارتند از جدول نماد (Symbol table) و در حال خطایم دار یا (error handler) زیرا در صورتی از شش نارطی ما میایم احتمال بروز خطا وجود ندارد در حال خطایم دار خطاهای احتمال را تا حد امکان برطرف می نماید که ما میایم برنامم متوقف نشود و برنامم نویسن برنامم خطا صادر نمرد.

COURCE Program



۱	۲	۳	۴	۵	۶
۷	۸	۹	۱۰	۱۱	۱۲
۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
۲۵	۲۶	۲۷	۲۸	۲۹	۳۰



کنون که بر کف گل چاه بناده صاف است

به صد هزار زبان بلبش در اوصاف است

و در بسیاری از عملیات بازنمایی و نوع آن ها مناسب
 با انواع دیگر برای نامگذاری آن در این جا باشد یا غیره می باشد

این دستور در زبان ++C به نوع لا به صورت float و نوع لا به صورت int در نظر
 گرفته شده و با توجه به این انواع و از آن گاهی که ملازمه عمل جمع و جمع و جمع اعتباری جدول
 سیستم کامپیوتری متفاوت می باشد این عمل قابل انجام خواهد بود تا به این نقطه می باشد
 می تواند انجام عمل جمع بر نام را متوقف نماید یا اینکه عمل تبدیل نوع انجام عمل تبدیل
 نوع می تواند به دو صورت محدود کننده و ارتقا یافته انجام شود اگر مثال فوق شوا داده لا را
 از نوع float به نوع int تبدیل کنیم محدود کننده خواهد بود که مطلوب نمی باشد زیرا فرض
 نماید x مقدار ۱۲.۴ و x = ۱۲.۴ قبل از دستورات به صورت ۱۲.۴ + دانسته که آنرا نخواهیم تبدیل
 نوع محدود کننده را به نوعی از اجزای عمل انجام از مقدار ۱۲.۴ به مقدار ۱۲ محدود می شود و ۰.۴
 از داده اولیه لا از بین خواهد رفت که ممکن است این ۰.۴ برای برنامه نویسی
 قابل ابرازش باشد و اگر تبدیل نوع نخواهد به عکس این حالت انجام دهد تبدیل نوع کمترش
 یا ارتقا یافته خواهد بود به معنی تبدیل y از int به float و خود عمل تبدیل نوع می تواند به دو صورت
 صریح و ضمنی انجام شود که تبدیل نوع صریح به شکل `int x = 12.4;` می باشد از پیش ساخته شده زبان
 توسط برنامه نویسی محدود و کمپایلر نوع صریح توسط فار تپلیر معنای ضیق
 از دید برنامه نویسی می توان صورت `int x = 12.4;` را به صورت `int x = 12;` ارتقا یافته خواهد بود
تولید کننده سینتکس یا مایکروسافت از این کار نامیده می شود به بعد برنامه نویسی معیار به تبدیل خواهد شد
 تبدیل از ارتقا یافته `int x = 12.4;` به `int x = 12;` صورت می گیرد در این کار نامیده می شود که معنای تولید می شود
 تا به نام نوشته شده با توجه به این که معنای در سطح معنای ما این برنامه معیار و تعیین معنای فاصله
 قرار می دهد.

۱	۲	۳	۴	۵	۶
۷	۸	۹	۱۰	۱۱	۱۲
۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
۲۵	۲۶	۲۷	۲۸	۲۹	۳۰



گل در برو می در کفا و معشوق به کار است

سلطان جهان به چنین روز غلام است

بهینه سازی کدها منظور از بهینه سازی است

الگوریتم بهینه سازی برای کدها می تواند به دو صورت زمان و حافظه باشد

در حافظه ای صورت بهینه سازی در حالت بهینه سازی زمانی یک کد معادل برای کدهای سطح

می شود یعنی که بدون تغییر عملکرد کد دارای تابع زمانی یا $T(n)$ مطلوب تر باشد

و در حالت بهینه سازی حافظه ای برای این کدها تعاضیم می دهیم که معادل یا بهینه تر بدون

تغییر عملکرد و صرف حافظه کمتری داشته باشند

تولید کدهای % این فاز از خروجی فاز بهینه سازی کد است که نتایج تولید می کنند که بعد از برای

کامپایلر ها که نتایج تولید شده زبان ماشین را همین خواهد بود

مثال: با فرض اینکه تغییرهای C, B, A هر عدد از یک صحیح سازد باشد خروجی شش ما را می بینیم
 $A_i = B + C * 10$

Lexical Analyser

تایپینگ

$id1 = id2 + id3 * 10$

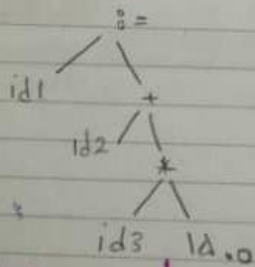
code optimization

$Temp1 = id3 * int_to_real(id)$

$id1 = id2 + temp1$

code generation

Syntax Analyser



intermediate code generation

$Temp1 = int_to_real(id)$

$Temp2 = id3 * temp1$

$Temp3 = id2 + Temp2$

$id1 = Temp3$

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		



لغو اعتبار کتابخانه به عنوان این است. میرزا حسن شیرازی، ۱۳۷۷ ه. ش

دری دیگر زدن اندیشه تبه دانست

به کوی میکه هر سالی که ره دانست