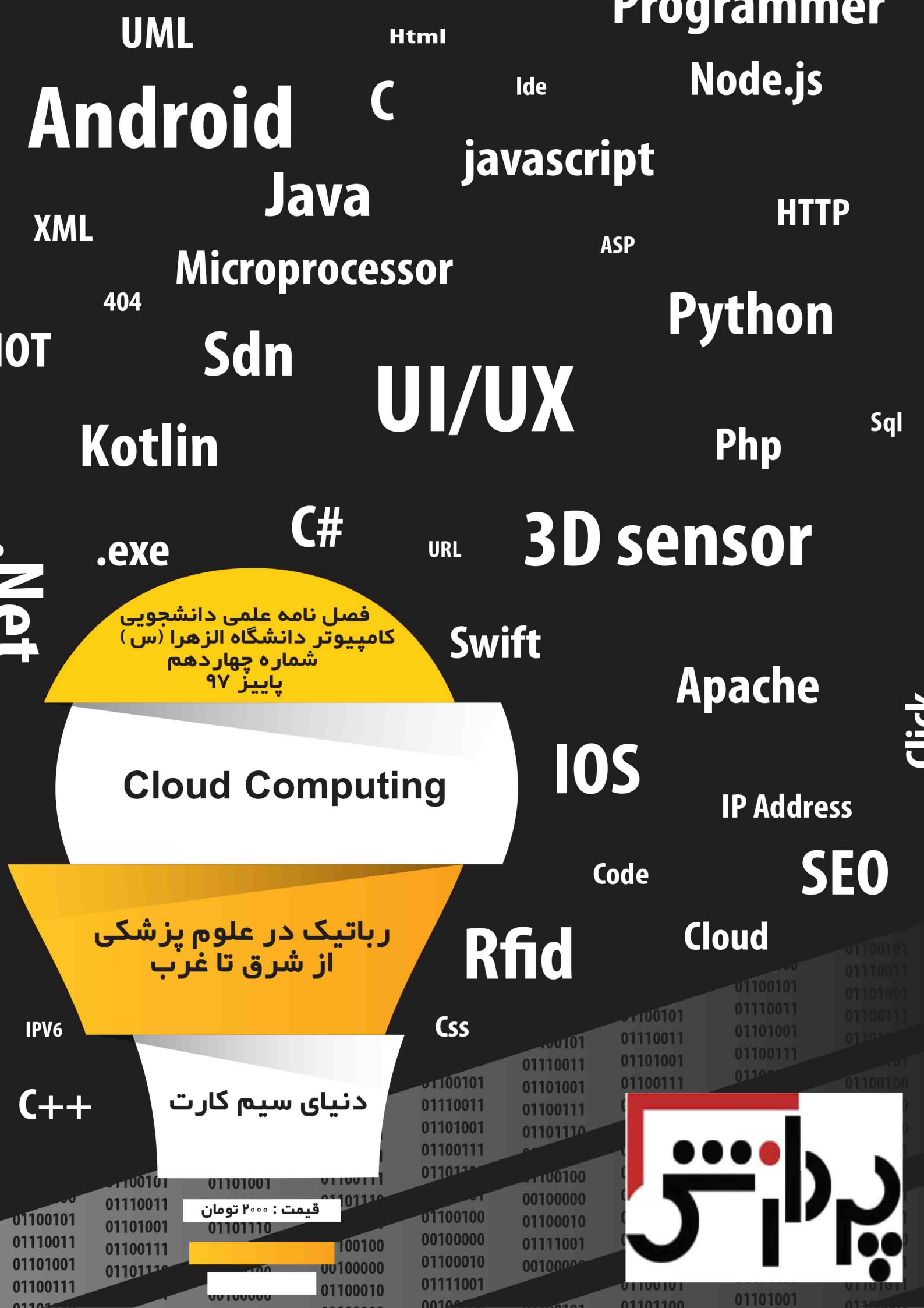

UML
Html
Programmer
Android
C
Ide
Node.js
javascript
Java
XML
HTTP
ASP
Microprocessor
404
Python
IOT
Sdn
UI/UX
Kotlin
Sql
Php
C#
.exe
URL
3D sensor
Net
فصل نامه علمی دانشجویی
کامپیوتر دانشگاه الزهرا (س)
شماره چهاردهم
پاییز ۹۷
Swift
Apache
Cloud Computing
IOS
IP Address
SEO
Code
رباتیک در علوم پزشکی
از شرق تا غرب
Rfid
Cloud
IPV6
Css
C++
دنیای سیم کارت
قیمت : ۲۰۰۰ تومان

پردازش

فصل نامه علمی - دانشجویی پردازش

پاییز 97

صاحب امتیاز: انجمن علمی کامپیوتر

کارشناس نشریه: زهرا وزیری

استاد مشاور: دکتر محمدرضا کیوان پور

مدیر مسئول: فاطمه زهرا ادیب

سردبیر: مبینا پاک

طراح جلد: ملیکا صباغیان

هیئت تحریریه: فاطمه زهرا ادیب - نسیم توحیدی - شبنم غلامشاهی - مبینا پاک - الهام بیدقی - زهرا قدسی - هما فراست - محمدحسن عباسی - میلاد اللّه یاری - نگار محبی - شیدا تارانی

ویراستاری: پریناز میرباقری - فاطمه اسدیار - فاطمه دومهری - پرستو جعفری

صفحه آرایی: سما کلانتری - ملیکا صباغیان - پریناز میرباقری - نازنین عباسی

مقدم - شیوا کرمی - پرستو جعفری

لیتوگرافی، چاپ و صحافی: چاپخانه دانشگاه الزهرا (س)

به نام خدا

همیشه زمانی که صحبت از سال تحصیلی جدید باشد، کلمات امید و اشتیاق در ذهن‌ام نقش می‌بندد... .

زیبایی آغاز سال تحصیلی از نظر من، دیدن دانشجویان ورودی جدید است. نه اینکه زبانم لال! چون سال آخری هستم آن‌ها را دست بیاندازم و به آن‌ها بخندم... ! مساله‌ی قابل توجه در مورد این عزیزان "اندک ترم" برای من، این است که ورودی‌های جدید، خوشحال یا ناراحت از نتایج‌شان در آزمون زندگی (=کنکور!)، حریصانه در جستجوی شاهراه پیشرفت هستند... . آن‌ها می‌دانند که بیل گیتس صرفا با قبول شدن در دانشگاه، "بیل گیتس" نشده... . (به علت بد آموزی اصل داستان را سانسور می‌کنیم ...). آن‌ها می‌خواهند "خفن" بشوند... . اما متاسفانه این سرمایه‌های فوق العاده، نمی‌دانند که چه راهی را باید طی کنند! به همین علت معمولا زمانی که ترم اولی‌ها در دانشکده رویت می‌شوند، سال بالایی بی نوایی را پیدا کرده‌اند و وی سوال پیچ می‌نمایند... . اما متاسفانه اکثرا به جای جواب، نصیحت‌هایی درباره‌ی اهمیت پاس کردن ریاضی یک و برنامه نویسی پیشرفته، دریافت می‌کنند (نصیحت ذکر شده را جدی بگیرید) ... .

در این شماره از نشریه‌ی پردازش، من و دوستانم سعی کردیم با درنظر گرفتن دوستان جدید و با عرض خیر مقدم به آن ها! مطالبی با موضوعات مناسب برای آن‌ها در نظر بگیریم تا شاید بتونیم در رسیدن به اهداف متعالی‌شان، کمک کوچکی باشیم ...

دوستان عزیز پردازشی جدید می‌توانید در صورتی که به مشاوره احتیاج داشتید و یا سوالی در ذهن‌تان بود، به ما ایمیل بزنید... . تا جایی که دانش‌اش را داشته باشیم به شما کمک می‌کنیم. اگر هم جواب را نداشتیم (ندانستن عیب نیست!) برایتان جواب یا راه رسیدن به آن را پیدا می‌کنیم. (جوینده یابنده است!)

ان شا ا... که به تمام اهداف زیبایتان برسید.. و "چه چیزی لذت بخش‌تر از از انجام دادن کاری که دیگران می‌گویند نمی‌توانی..."

سال تحصیلی سرشار از شادی و موفقیت را برای شما دانشجویان عزیز ، اساتید بزرگوار و کارمندان گرامی آرزومندیم

هم شاگردی سلام ...

مهر ماه ۹۷

مبینا پاک

رایانش ابری دیگه چیه؟؟!!!

امروزه اکثریت مردم در سراسر جهان بارها و بارها کلمه «Cloud computing»را می‌شنوند، اما بسیار اندک‌اند افرادی که معنای واقعی این اصطلاح را بدانند. در این مطلب سعی می‌کنیم معنی آن را به نحوی ساده‌تر و قابل‌فهم‌تر بیان کنیم.

Cloud computing چیست؟

این اصطلاح که در فارسی بدان رایانش ابری گویند بر اساس تعاریف موسسه ملی فناوری و استانداردها، اینگونه تعریف شده: رایانش ابری مدلی است که برحسب تقاضای شبکه، دسترسی آسان و فراگیر به مجموعه عظیمی از منابع محاسباتی قابل تنظیم (همانند شبکه‌ها، سرورها، فضای ذخیره سازی، برنامه‌های کاربردی و سرویس‌ها) را به سرعت و بدون دخالت سرویس‌دهنده به راحتی ممکن می‌سازد.

البته این تعریف بسیار کلی و تا حدودی گیج کننده است. *پس تعریف ساده‌تر آن برای افراد عادی چیست؟*

رایانش ابری = نرم افزارهای تحت وب به همین راحتی!

اگر شما از نرم‌افزارهای تحت‌وب که توسط سرویس‌دهندگانی همچون گوگل و مایکروسافت ارائه شده استفاده کنید، در واقع از رایانش ابری بهره مند می‌شوید. هر کدام از اپلیکیشن‌های مبتنی بر وب از جمله Gmail ، Google Calendar، Hotmail، Dropbox و Google Doc برمبنای رایانش ابری تعریف شده‌اند. زمانی که شما به هرکدام از این سرویس‌ها متصل می شوید در واقع به منبع گسترده‌ای از سرورهای موجود در اینترنت متصل شده‌اید. نیاز نیست که کاربر حتما از مرورگر وب برای رایانش ابری استفاده کند؛ اما ظاهراً همه چیز در حال حرکت به این سمت و سو است. برای مثال اگر از سرویس Dropbox استفاده می کنید کمتر پیش می آید که از مرورگرتان برای انتقال فایل ها استفاده کنید. اما در برای دسترسی جیمیل اغلب از مرورگر استفاده می کنید.

چرا Cloud Computing؟

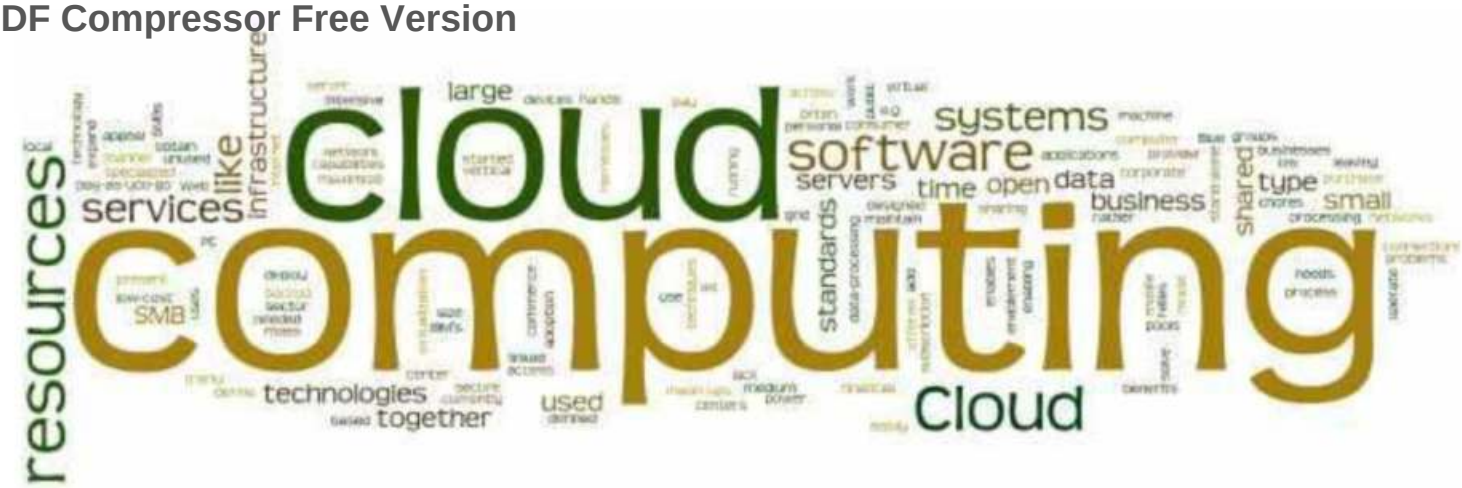
همانگونه که گفتیم این اصطلاح نامفهوم که در واقع نرم‌افزارهای تحت وب را توصیف می‌کند؛ مدت‌هاست که در اطراف ما وجود داشته اما بیشتر افراد آن را به این اسم نمی‌شناختند. پس به منظور رسیدن این بازار به شروعی جدید و جایگزینی اپلیکیشن‌های تحت وب به جای نرم‌افزارهای وابسته به سرورهای میزبان، این اصطلاح به وجود آمده است.

توصیف اینکه چرا از کلمه «Cloud» یا «ابر» در این اصطلاح استفاده شده نیز بسیار ساده است. اگر توجه کرده باشید در نمودارهای شبکه‌های رایانه‌ای، اینترنت معمولاً به شکل ابر در تصویر نمایش داده می‌شود. شاید دلیل این تشبیه این است که اینترنت همانند یک ابر، جزئیات فنی‌اش را از دید کاربران مخفی نگه می‌دارد. بنابراین در واقع این عبارت راهی برای مشاوران و شرکت‌ها فراهم آورده تا خدمات خود را در بسته بندی‌های جدید به فروش بگذارند.

رایانش ابری چگونه به ما کمک می کند؟

از آنجا که شرکت‌ها و برندهای تجاری در حال انتقال نرم‌افزارهای خود بر روی وب هستند و این اپلیکیشن‌ها روز به روز با ویژگی‌های جدید و جالب‌تری از طریق مرورگرها به معرض نمایش در می‌آیند؛ می‌توان گفت: به زودی قادر خواهیم بود که از هر مرورگر و با هر کامپیوتر و بدون وجود هیچگونه مرزی بین کامپیوتر شخصی‌مان و اینترنت به همه چیز دست پیدا کنیم.

سرویس Dropbox مثال عالی از رایانش ابری است که هم اکنون میلیون‌ها نفر از آن استفاده می‌کنند تا همیشه به فایل‌های مهم‌شان دسترسی



دسترسی داشته باشند. با ثبت‌نام در این سایت و نصب یک نرم‌افزار روی کامپیوترتان، شما پوشه‌ای در کامپیوتر خواهید داشت که هر چیزی درون آن قرار بدهید با اکانت آنلاین شما هم بروز رسانی می‌شود. حالا می‌توانید دراپ باکس را روی کامپیوتر دوم‌تان هم نصب کنید. حالا هر فایلی درون این پوشه در یک کامپیوتر قرار بگیرد روی دومی هم ظاهر خواهد شد. علاوه بر این، یک نسخه از فایل‌هایتان هم به صورت امن روی اینترنت قرار دارد. با استفاده از این سرویس دیگر محدود به یک کامپیوتر نیستید و اطلاعات شما در ابر بزرگ اینترنت قرار دارد و همیشه و از هر کجا به آن دسترسی دارید. ضمن اینکه می‌توانید برخی پوشه‌ها را هم با دوستانتان به اشتراک بگذارید. در کنار همه اینها امکان دسترسی به فایل‌هایتان از طریق تلفن همراه را نیز خواهید داشت. حالا که مایکروسافت نسخه بتای Internet Explorer 9 را ارائه کرده که از استانداردهای جدید وب همانند HTML5 پشتیبانی می‌کند و نیز از شتاب‌دهنده سخت‌افزاری برای افزایش سرعت مرورگر بهره می‌برد؛ بالاخره تمامی مرورگرها در یک پایه قرار گرفتند. پس زمانی که مایکروسافت اعلام نمود که IE9 برای تغییر آمده؛ شوخی نبود چرا که واقعاً این تغییر را در مقایسه با مرورگرهای IE7 ، IE8 و IE6 کاملاً احساس می‌کنیم. البته داستان از زمانی جذاب‌تر شد که سیستم عامل Chrome نیز نهایتاً به بازار عرضه گردید که اساساً سیستم عاملی مبتنی بر وب است.

رایانش ابری در زندگی ما چه تغییراتی ایجاد می‌کند؟

در گذشته کمپانی‌های مختلف، تمامی نرم‌افزارهای خود را بر روی سرورها و هاست خویش اداره می‌کردند و واضح است که این امر نیاز به صرف هزینه و حفظ و نگهداری شرایط در حالت ایده‌آل دارد. از دیدگاه تجاری، می‌توان اینگونه گفت که این سرویس‌ها هم اکنون می‌توانند بسیاری از محاسبات خود را به سرویس‌های ابری انتقال دهد و اپلیکیشنی را که می‌بایست بر روی سرورهای خود نصب کنند به راحتی و از طریق اینترنت در اختیار مشتریان قرار دهند. آیا شما در مورد شرکت‌هایی که به Google Doc سوئیچ کرده اند چیزی شنیده‌اید؟ این مثال نمونه کاملی از سوئیچ شرکت‌ها از میزبانی سرورهای محلی خود به استفاده از رایانش ابری است.

اما اگر شرکت شما بخواهد سرویسی را برای مشتریان فراهم کند چطور؟ شما می‌توانید امکان استفاده از رایانش ابری را با ساخت نرم‌افزارهایی که نیاز به نصب بر روی سرورتان ندارند اما از منابع سرور استفاده می‌کنند فراهم کنید. شرکت‌های بزرگی همچون گوگل، مایکروسافت و آمازون امکاناتی ساخته‌اند که چنین خدماتی را به شما ارائه می‌دهند. در سرویس‌دهی جدید شما برای سرور هزینه‌ای پرداخت نمی‌کنید بلکه پرداخت شما برای CPU و دیگر منابعی است که نرم‌افزارتان از آن استفاده می‌کند، بنابراین مخارجتان کمتر خواهد شد و در هنگام نیاز می‌توانید به سرعت منابع مورد نیاز خود را ارتقا دهید.

رایانش ابری دیگه چه کاربردهایی داره ؟

- **تجارت و کسب و کار:**

انسان‌ها برای کسب درآمد و برآورده‌سازی نیازهای خود، به زمینه‌های شغلی مختلفی روی می‌آورند و در این مسیر از ابزارهای مختلفی برای بهینه‌سازی و افزایش بهره‌وری در کسب و کار خود سود می‌برند. بدیهی است که رایانش ابری نیز از این مسئله مستثنی نیست و اغلب اهالی دنیای کسب و کار، پتانسیل‌های این فناوری در زمینه‌ی تجارت را دریافته‌اند. در مطالعه‌ای که با همکاری نشریه‌ی اکونومیست و شرکت آی‌بی‌ام انجام شده، از میان ۵۷۲ کسب و کار مورد بررسی، نزدیک به سه‌چهارم آن‌ها در حال استفاده از رایانش ابری یا آزمایش این تکنولوژی هستند. علاوه‌بر این، احتمال می‌رود ۹۰ درصد از این کسب و کارها در سه سال آینده به مشتریان رایانش ابری تبدیل شوند.

دلیل استقبال عمده‌ی کسب و کارها از رایانش ابری، مزایایی است که این فناوری با خود به‌همراه دارد. یکی از این مزایا انعطاف‌پذیری رایانش ابری است که به کاربران اجازه می‌دهد در هر زمان به تناسب نیاز خود منابعی را خریداری کنند؛ همین مسئله به کاربران کمک می‌کند تا از هزینه‌های اضافی برای منابعی که نیاز چندانی به آن‌ها ندارند خلاص شوند. یکی از مثال‌های قابل توجه در این زمینه، سرویس استریم ویدئوی نتفلیکس است؛ این شرکت از سرویس‌های ابری ***آمازون*** استفاده می‌کند. نتفلیکس در ساعات اوج استفاده‌ی تماشاگران از سرویس خود، منابع ابری بیشتری را مورد استفاده قرار می‌دهد و در ساعات دیگر از منابع کمتری بهره‌برداری می‌کند.

تصور کنید به‌جای استفاده از سرویس‌های آمازون، نتفلیکس از دیتاسنترهای اختصاصی خود استفاده می‌کرد؛ نه‌تنها این شرکت مجبور بود هزینه‌ی زیادی برای ایجاد و نگهداری از این دیتاسنترها صرف کند، بلکه همچنین در ساعات کاهش تعداد تماشاگران، بخش زیادی از توانایی این دیتاسنترها بلااستفاده می‌ماند. تطبیق‌پذیری نیز مزیت دیگر رایانش ابری است که به سرویس‌هایی نظیر نتفلیکس و Active video اجازه می‌دهد محتوای خود را در دستگاه‌های متخلف نظیر رایانه‌های شخصی، تلفن‌های همراه و... ارائه دهند.

- **پزشکی:**

در سال‌های اخیر، رایانش ابری با سرعت قابل توجهی به حوزه‌ی پزشکی نیز وارد شده است. منابع نامحدود ارائه شده توسط رایانش ابری و انعطاف‌پذیری ذاتی این فناوری، موجب شده تا از این فناوری برای توسعه و ارائه‌ی خدمات درمانی استفاده شود. علاوه‌بر این، از قدرت پردازش ابری در مطالعات پزشکی در زمینه‌هایی نظیر ژنتیک و پزشکی مولکولی نیز استفاده می‌شود.

ذخیره‌سازی اطلاعات بیماران به‌طور محلی، چه به‌صورت آرشیوهای فیزیکی قدیمی و چه به‌صورت الکترونیکی و در سرورهای داخلی، نه‌تنها هزینه‌بر است، بلکه پیچیدگی‌های خاصی را نیز برای مراکز پزشکی به‌همراه می‌آورد. به‌همین دلیل بیمارستان‌ها و مراکز پزشکی نیز در حال روی آوردن به ذخیره‌سازی ابری اطلاعات بیماران هستند، این مسئله باعث می‌شود پزشکان و بیماران بتوانند در هر لحظه و بدون مراجعه به مراکز پزشکی، به اطلاعات مورد نیاز خود دسترسی داشته باشند.

یکی از مثال‌های موفق در این زمینه، شبکه‌ای موسوم به Health Hi Wayدر هندوستان است. این شبکه با متصل کردن بیش از ۱۱۰۰ بیمارستان و ۱۰ هزار پزشک به یک‌دیگر، نه‌تنها اشتراک اطلاعات و انجام هماهنگی‌های مختلف میان مدیران، پزشکان و شرکت‌های فعال در حوزه‌ی پزشکی را ساده کرده، بلکه هزینه‌های پزشکی را نیز به‌مراتب کاهش داده است.

- **آموزش:**

حوزه‌ی آموزش نیز همچون دیگر حوزه‌ها از مزایای رایانش ابری بی‌بهره نمانده است. رایانش ابری انتخابی مناسب برای آن‌دسته از موسسات آموزشی است که فاقد توان مالی کافی برای تهیه و نگه‌داری زیرساخت‌های اطلاعاتی هستند. علاوه‌بر این، به‌کمک رایانش ابری، دسترسی به خدمات آموزشی به‌صورت راه‌دور نیز آسان‌تر از گذشته شده‌است.

می‌توان از افزایش بهره‌وری زمانی، کاهش هزینه‌ها و ساده‌شدن رویه‌های آموزشی به‌عنوان مزایای اصلی رایانش ابری در حوزه‌ی آموزش یاد کرد. این مزایا، به‌ویژه برای کشورهایی که با چالش‌های بودجه‌ای روبه‌رو هستند، بسیار قابل توجه خواهد بود.

معماری رایانش ابری

در حالت کلی و جامع ترین نوع معماری، رایانش ابری شامل ۵ لایه اصلی است.

۱) **کاربر**

۲) **نرم افزار به عنوان سرویس**

۳) **بستر به عنوان سرویس**

۴) **زیرساخت به عنوان سرویس**

۵) **سرور**

این ۵ لایه ، **معماری رایانش ابری** را تشکیل می‌دهند که در ادامه به توضیح و مثال هر کدام خواهیم پرداخت.

- **کاربر:**

کاربر یا **Client** در معماری پردازش ابری به معنای یوزر و یا فردی که از **Cloud** استفاده می‌کند، نیست.

کاربر در **رایانش ابری** متشکل از سخت افزار یا نرم افزاری است که برای تحویل برنامه‌های ابر فعالیت می‌کند و به طور ویژه وظیفه رساندن سرویس به دست یوزر را دارد.

ویژگی های کاربر و نحوه شناسایی آن

به طور کلی به نرم افزار و سخت افزاری که به تنهایی کارایی نداشته و صرفاً یک سخت افزار یا نرم افزار خاموش است، کاربر گفته می‌شود.

از جمله مثال‌هایی که می‌توان برای کاربر نام برد:

۱) **گوگل کروم** (مرورگر کروم بدون اتصال به ابر گوگل کاملا بی استفاده خواهد بود)

۲) **نرم افزار گوگل درایو** (بدون اتصال به ابر گوگل درایو بدون استفاده است)

۳) **نرم افزار One Drive** (نرم افزار متصل شدن به فضای ابری مایکروسافت که بدون اتصال بی‌مصرف است)

در این بخش مثال‌های زیادی را می‌توان ذکر کرد که با شناختن نرم افزارهای کاربری بالا، شما می‌توانید سایر نرم افزارهای این لایه را شناسایی کنید.

- **نرم افزار به عنوان سرویس:**

لایه بعدی لایه ایست با نام **نرم افزار به عنوان سرویس** که در انگلیسی آن را **Software as a Service** هم می‌خوانند و مخفف آن نیز **SaaS** است.

SaaS، نرم افزارها را به صورت یک سرویس بر روی اینترنت تحویل داده و بدین وسیله کاربر می‌تواند به آن متصل شده و از آن بهره ببرد.

بدین شکل، سرویس دهی نرم افزار بر روی ابر بسیار آسان بوده و آپدیت، نگه‌داری و پشتیبانی به صورت متمرکز تنها بر روی ابر صورت می‌گیرد.

از ویژگی‌های اصلی نرم افزار به عنوان سرویس می‌توان به

موارد زیر اشاره کرد:

- تحویل نرم افزار به صورت مدل یک به چند (در این حالت یک نرم افزار در حال اجرا است و چندین کاربر از آن استفاده می‌کنند.)
- بروزرسانی و ارتقای نرم افزاری به صورت متمرکز بوده و نیازی نیست برای تمامی کاربران آپدیت و یا ارتقا صورت پذیرد.
- دسترسی و مدیریت نرم افزارهای تجاری

مثال SaaS :

Google Labs: یک آزمایشگاه برای تست‌های نرم افزاری متفاوت است.

Salesforce.com: یکی از گسترش‌دهندگان سیستم رایانش ابری است که نرم افزارهای سازمانی و دولتی تحت سرور را می‌تواند در بستر اینترنت در دسترس کاربران قرار دهد.

Facebook: فیسبوک برای سرویس دهی از یک اینترفیس اینترنتی استفاده می‌کند اما در اصل یک نرم افزار است که در حال اجرا شدن می‌باشد.

You tube: سایت یوتیوب در اصل یک نرم افزار قدرتمند است که در پس پرده فعالیت می‌کند و شما تنها لایه کاربر آن را می‌بینید که به صورت صفحه وب به نمایش در می‌آید.

- **بستر به عنوان سرویس:**

بستر به عنوان سرویس لایه ایست که در انگلیسی آن را **PaaS** یا **Platform as a Service** می‌خوانند.

لایه **PaaS**، قسمتی از **معماری رایانش ابری** است که نرم افزار های لایه **SaaS** روی آن نصب می‌گردد و اجرای آن به عهده این لایه می‌باشد.

در تعریفی ساده می‌توان گفت این لایه، نرم افزارهای لایه **SaaS** را به عنوان یک سرویس بر روی بستر شبکه (اینترنت) ارائه می‌دهد.

بستر به عنوان سرویس (PaaS) در اصل یک بسته نرم افزاری جامع است که امکاناتی مانند تجمیع، تبادل پیغام، اطلاعات و میان افزار است که ارتباط بین لایه **SaaS** و **IaaS** را برقرار می‌کند.

مثال PaaS :

موتور نرم افزار Google Apps: این ابزار امکان اجرای برنامه‌های کاربردی توسط زیر ساخت‌های گوگل را می‌دهد.

پلتفرم‌هایی مانند **Google Apps** می‌توانند امکانات پایه قدرتمندی را در اختیار توسعه دهندگان نرم افزارها قرار دهند اما افزایش امکانات زیرساختی و ... در دست گوگل است که آن‌ها را افزایش دهد یا خیر.

سیستم عامل Azure: سیستم عامل **Azure** مایکروسافت یک **PaaS** است که می‌توان از آن برای توسعه نرم افزارها در سطح اینترنت استفاده نمود.

- **زیرساخت به عنوان سرویس:**

لایه چهارم از معماری رایانش ابری است که به آن **IaaS** و یا **Infrastructure as a Service** نیز می‌گویند.

سرویس زیرساخت **رایانش ابری** در اصل یک بستر مجازی است که به صورت سرویس ارائه می‌شود. کاربران به جای خرید سخت‌افزار، نرم‌افزار، فضای دیتاسنتر و یا تجهیزات شبکه می‌توانند تمام آنها را به صورت یک سرویس مجازی خریداری کرده و از آن استفاده کنند.

هزینه‌های پرداختی کاربران برای این سرویس عموما به شکل رایانش همگانی و میزان منابع مصرف شده دریافت می‌گردد و هزینه پرداختی معادل هزینه ایست که سرویس شما فعالیت داشته است.

ارائه این لایه به مشتریان در اصل مدل پیشرفته و تکامل یافته عرضه سرورهای مجازی خصوصی است.

در مجموع می‌توان با **IaaS**، زیر ساخت کامپیوتر و محیط پلتفرم مجازی سازی را مانند یک سرویس در اختیار کاربران قرار داد.

مثال IaaS :

از نمونه‌هایی که امروزه می‌توان به آن اشاره کرد سرویس **Amazon EC2**، **Azure Platform**، **Sun Parascale** می‌باشد.

- **سرور:**

لایه سرور به سخت افزار و نرم افزارهایی اطلاق می‌شود که به صورت کامل وظیفه پردازش را در ابر به عهده دارند.

به عنوان نمونه می توان به پردازنده‌های چند هسته‌ای موازی و سوپر سرورهای پایه ابر اشاره کرد.

openstack چیست؟

اوپن استک (به انگلیسی: OpenStack) یک پلت‌فرم نرم‌افزاری متن باز برای رایانش ابری است که توسط Rackspace و ناسا در ژوئیه ۲۰۱۰ طرح‌ریزی شد. هم اکنون بیش از ۵۰۰ شرکت از جمله IBM ، فوجیتسو، اوراکل، یاهو، سیتریکس، دل، AMD، اینتل، کنونیکال، سوزه، اچ‌پی و سیسکو سیستمز و شرکت‌های بسیار دیگری به این پروژه پیوسته‌اند.

OpenStack یک پروژه رایانش ابری است که به تجهیز بستری متن باز و قابل دسترس در همه جا برای ابرهای شخصی و عمومی کمک می‌کند. این پروژه توسط OpenStack Foundation مدیریت می‌شود که یک بنیاد غیرانتفاعی است که در سپتامبر ۲۰۱۲ تأسیس شده است.

اپن استک به کاربران اجازه می‌دهد تا بتوانند ماشین‌های مجازی و دیگر نمونه‌ها برای کنترل وظایف گوناگون در مدیریت یک محیط ابری در حال اجرا را توسعه دهند. این مجموعه ابزار گسترش عمودی را آسان می‌کند به این معنی که وظایفی که از اجرای همزمان سود می‌برند، می‌توانند در زمان اجرا، به آسانی با تنظیم نمونه‌های متفاوت، به کاربران بیشتر یا کمتری خدمات ارائه دهند. برای مثال، یک اپلیکیشن موبایل، که باید با یک سرور از راه دور ارتباط برقرار کند، می‌تواند کار ارتباطی خود را با کاربران دیگر در طول نمونه‌های مختلف، تقسیم کند که همه اینها با سرعت و به آسانی با یکدیگر در ارتباط خواهند بود؛ و مهمتر از آن، اپن استک یک ابزار متن باز است، به این معنی که هر فردی که آن را انتخاب کند، می‌تواند به کد منبع آن نیز دسترسی داشته، هر تغییر یا اصلاحی را که مورد نیاز است، اعمال کند و به صورت رایگان، تغییرات اعمال شده را در سطح وسیعتری به اشتراک بگذارد. متن بازبودن آن نیز این امکان را هم می‌دهد که هزاران توسعه دهنده در سرتاسر جهان می‌توانند بر روی آن کار کنند و مجموعه نرم‌افزاری با ویژگی‌های قوی تر و امن‌تر تولید کنند.

نحوه کار با اپن استک چیست ؟

رایانش ابری یا Cloud در واقع فراهم کردن محیط محاسباتی برای کاربران پایانی از راه دور است، به صورتی که نرم‌افزارها به عنوان یک سرویس بجای اینکه بر روی کامپیوترهای خود کاربران اجرا شوند، بر روی سرورهای قابل اعتماد و توسعه پذیر اجرا می‌شوند. محاسبات ابری می‌تواند چیزهای دیگری را هم شامل شود ولی معمولاً منظور اصلی، اجرای آیتم‌های مختلف مانند نرم‌افزارها، پلتفرم‌ها و زیرساخت، به عنوان یک سرویس می‌باشد. اپن استک در واقع جزء دسته‌بندی سوم می‌شود و یک زیرساخت به عنوان سرویس یا [IaaS]محسوب می‌شود. فراهم کردن زیرساخت به این معنی است که کاربران با استفاده از اپن استک، به آسانی می‌توانند نمونه‌های جدید اضافه کرده و مولفه‌های دیگر cloudرا بر روی آنها اجرا کنند. معمولاً، این زیرساخت یک پلتفرم را اجرا می‌کند که توسعه دهنده بر روی آن می‌تواند اپلیکیشن‌های نرم‌افزاری را ایجاد کرده و به کاربران پایانی خود خدمات ارائه دهد

سرویس های open stack چیست؟

اپن‌استک از قسمت‌های اجرایی مختلفی ساخته شده است. بخاطر ذات آزاد بودن آن، هر شخصی می‌تواند مولفه‌های دیگری نیز به آن بیفزاید تا نیازهای خود را برآورده سازد. انجمن اپن استک، نه مولفه کلیدی را مشخص کرده است که در واقع بخشی از هسته اپن استک محسوب می‌شوند و به عنوان بخشی از هر سیستم اپن استکی توزیع می‌شوند و به صورت رسمی توسط انجمن اپن استک، پشتیبانی می‌شوند.

Nova:

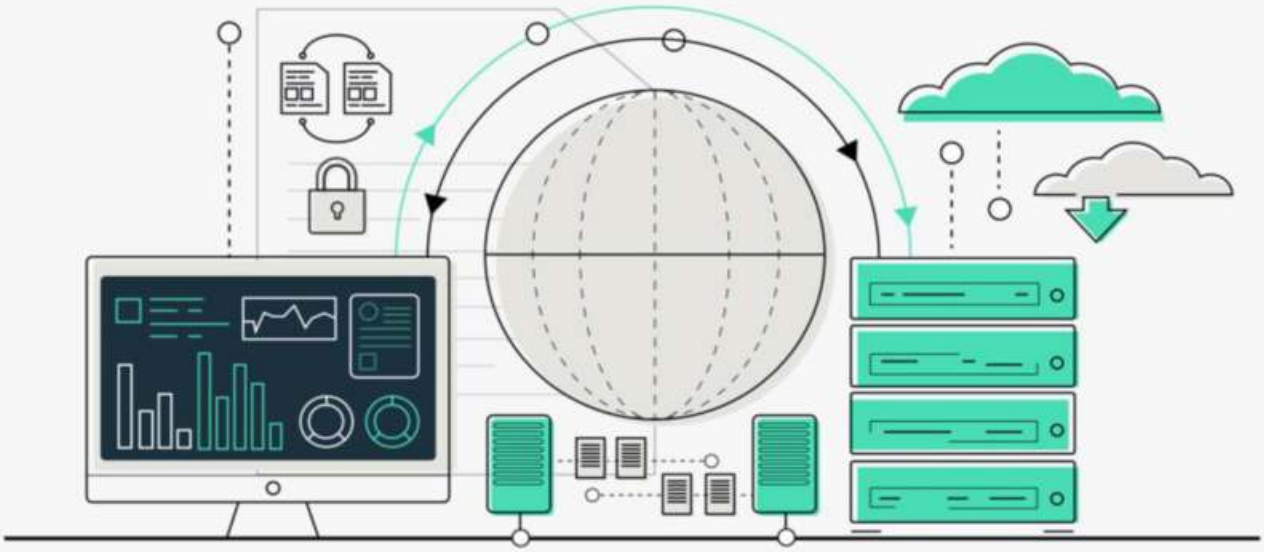
Nova، موتور محاسباتی اصلی، برای اپن استک است. این موتور برای توسعه و مدیریت تعداد زیادی از ماشین‌های مجازی و نمونه‌های دیگر برای کنترل وظایف محاسباتی استفاده می‌شود.

Swift:

سوئیفت، یک سیستم ذخیره‌سازی اشیا و فایل‌هاست. بجای ایده قدیمی که برای بدست آوردن فایل‌ها به آدرس آنها بر روی دیسک ارجاع می‌داد، توسعه دهندگان می‌توانند به جای ارجاع به یک آدرس منحصربفرد، به یک فایل یا بخشی از اطلاعات ارجاع دهند و به اپن استک این فرصت را بدهند که آنها را در هر جایی ذخیره کند. این کار، توسعه بخشی را آسان می‌کند و در این صورت توسعه دهندگان دغدغه ظرفیت یک سیستم منفرد ورای نرم‌افزار را نخواهند داشت.

Cinder:

Cinder یک مولفه ذخیره‌سازی بلاک است که شباهت بیشتری به مفهوم قدیمی دسترسی به مکان‌های خاص یک دیسک درایو دارد. این



روش سنتی برای دسترسی به فایل‌ها معمولاً به خاطر اهمیت سرعت دسترسی به داده، ممکن است مورد بحث باشد.

Neutron:

Neutron قابلیت شبکه سازی برای اپن استک را فراهم می‌کند. این ویژگی به هر مولفه یک پلتفرم اپن استک کمک می‌کند تا بتواند با دیگری به صورت کارا و سریع ارتباط برقرار کند.

Horizon:

یک پنل کنترلی برای اپن استک است. این پنل تنها واسط گرافیکی اپن استک است که در واقع می‌تواند اولین مولفه برای کاربرانی باشد که می‌خواهند اپن استک را تست کنند. توسعه دهندگان می‌توانند به تمام مولفه‌های اپن استک از طریق API دسترسی پیدا کنند ولی این پنل کنترلی شرایطی برای مدیران سیستم فراهم می‌کند تا نگاهی اجمالی به کل سیستم cloudداشته باشند و در صورت نیاز آن را مدیریت کنند.

Keystone:

این ابزار در واقع سرویس‌های هویت را برای اپن استک فراهم می‌کند. در اینجا لیستی از کاربران اپن استک به همراه دسترسی‌های مجاز آنها نگهداری می‌شود. این سرویس، ابزارهای دسترسی چندگانه فراهم می‌کند به این معنی که توسعه دهندگان می‌توانند به آسانی دسترسی کاربر موجود را با استفاده از روش‌هایی در keystone نگاشت کنند.

Glance:

وظیفه Glance، ارائه سرویس image در اپن استک است. در این مورد، image ها به image های موجود در هارد دیسک ارجاع داده می‌شوند. این image ها می‌توانند در زمان توسعه یک نمونه ماشین مجازی جدید، به عنوان الگو استفاده شوند.

Ceilometer:

Ceilometer، سرویس تله متری را برای اپن استک فراهم می‌کند که در واقع فراهم کننده سرویس‌های صورتحساب برای کاربران فردی محاسبات ابری است. این سرویس نیز، میزان قابل تأییدی از استفاده کاربر از هر کدام از مولفه‌های یک اپن استک را نگهداری می‌کند.

Heat:

Heatدر واقع مولفه هماهنگ سازی و تنظیمات در اپن استک است که به توسعه دهندگان این امکان را می‌دهد که نیازهای اپلیکیشن‌های cloud را که منابع لازم برای هر برنامه را تعریف می‌کند، در یک فایل ذخیره کنند. با این روش، می‌توان زیرساخت موردنیاز را که سرویس cloud باید بر روی آن اجرا شود، مدیریت کرد.

Mistral:

Mistral خدمتی است که گردش‌های کاری را مدیریت می‌کند. به طور معمول کاربر با استفاده از زبان‌هایی مانند YAML که بر پایه‌ی گردش کاری کار می‌کنند، یک گردش کاری را می‌نویسد و تعریف گردش کاری مذکور را با استفاده از رابط برنامه‌نویسی کاربردی بر پایه REST برای Mistral ارسال می‌کند. سپس کاربر می‌تواند به صورت دستی یا با استفاده از پیکربندی یک یا چند رویداد، گردش کاری را فعال نماید.

Trove:

Trove خدمت Database-as-a-Service را به عنوان موتور پایگاه‌داده رابطه‌ای) بر پایه (SQL و همچنین غیر رابطه‌ای (NoSQL) را برای اپن‌استک فراهم می‌کند.

اپلیکیشن های شبه کلود

حتی فکر کردن به اینکه شما در یک اتفاق تمام اطلاعات خود را از دست بدهید نیز ناگوار خواهد بود، به خصوص اگر شما از این اطلاعات پشتیبانی نیز تهیه نکرده باشید، در این جا اهمیت تکنولوژی ذخیره سازی ابری معلوم می‌شود.

به یاد می‌آورم که یک بار در یوتیوب ویدیویی را دیدم که در آن یک مجری از شهروندان آمریکایی سوال می‌کرد که آیا وضعیت آب و هوا بر روی اطلاعات آنان در سرویس‌های ابری تاثیر می‌گذارد؟ پاسخ بیشتر آمریکایی‌ها (که به پایین بودن اطلاعات عمومی‌شان معروفند) مثبت بود! اما فضای ابری در واقع یک سرویس ارائه فضای ذخیره‌سازی به کاربران است تا بتوانند اطلاعات خود را بر بستر اینترنت ذخیره‌سازی کنند تا هم خیالشان از بابت خراب شدن حافظه و از بین رفتن اطلاعات‌شان راحت شود و هم این که بتوانند از هر کجا و به وسیله گجت‌های مختلف به اطلاعات خود دسترسی داشته باشند.

در این تکنولوژی شما صد در صد از سلامتی داده‌های خود اطمینان خواهید داشت و داده‌های موبایل که قبلا جزو ناامن‌ترین‌ها به شمار می‌رفتند، اکنون با تمرکز ذخیره‌سازی ابری بر روی آنها می‌توانند از ایمنی و پشتیبانی مطمئنی برخوردار باشند. در این مقاله ۵ تا از بهترین سرویس‌های ذخیره‌سازی ابری را معرفی خواهیم کرد.

- **گوگل درایو**

این سیستم را می‌توان از بهترین نمونه‌های فضای ابری دانست که توسط گوگل پشتیبانی می‌شود. می‌توان در آن به طور همزمان به ذخیره سازی و اشتراک گذاری اطلاعات و فایل‌ها پرداخت. شما نیاز به ثبت‌نام یا ساختن اکانت‌های متعدد نخواهید داشت چرا که این سیستم برروی حساب گوگل شما فعال خواهد بود. در سرویس رایگان آن شما می‌توانید تا ۵ گیگابایت ذخیره سازی را انجام بدهید و در سرویس پولی این میزان تا ۱۰۰ گیگابایت افزایش خواهد یافت. سرعت آپلود و راحتی کار با آن از دیگر مزایای این سیستم است.

- **وان درایو**

این سرویس هم برای افرادی مناسب است که خواهان سرعت بالا در آپلود هستند و محیط آن مانند اینترنت اکسپلور است. تا ۲۵ گیگابایت قابلیت ذخیره سازی دارد و در آن تا ۵۰ مگابایت به طور همزمان می‌توان فایل ذخیره کرد. به این سیستم از طریق وب‌سایت می‌توان دسترسی پیدا کرد و اپلیکیشن ویندوزفون، آیفون و اندروید نیز برای آن منتشر شده است. همچنین قابلیت ویرایش فایل‌های آفیس به صورت مستقیم در آن برقرار است.

- **باکس**

در باکس متناسب با نوع ثبت نام بین ۵ تا ۵۰ گیگابایت فایل به صورت رایگان می‌توان ذخیره کرد و قابلیت مهم آن راحتی درگ و دراپ کردن فایل‌ها از روی دسکتاپ برای آپلود است. اپلیکیشن‌های اندروید، آیفون، آیپد و بلک بری برای باکس وجود دارند تا در هر کجا که هستید بتوانید به فایل‌هایتان دسترسی داشته باشید. قابلیت ویرایش مستقیم فایل‌های ورد و اکسل نیز به کمک افزونه‌هایی در آن وجود دارد.

- **ای درایو**

ADrive از معدود سرویس‌های ابری است که در آن می‌توانید تا ۵۰ گیگابایت ذخیره سازی کنید. شما می‌توانید به راحتی با درگ و دراپ کردن فایل‌ها را آپلود کنید و با

نرم‌افزار مخصوص خود سایت، ویرایش را بر روی فایل‌ها انجام بدهید. این سیستم قابلیت ذخیره‌سازی و پشتیبانی تمامی فایل‌ها از جمله موسیقی، ویدیو، عکس و... را دارد.

- **آی کلود**

این سرویس برای دستگاه‌های با سیستم عامل iOS موجود است و به شما قابلیت ذخیره‌سازی تا ۵ گیگابایت را می‌دهد و اگر مورد نیاز شما نبود باید برای افزایش آن مبلغی را بپردازید. تمام کسانی که فایلی را از فروشگاه iTunes خریداری کرده و از طریق این سیستم اشتراک گذاری می‌کنند، از حجم ۵ گیگابایتی شان کم نمی‌شود. این سیستم نیز مانند سرویس قبلی قابلیت ذخیره سازی همه نوع فایل را به شما می‌دهد.

- **دراپ باکس**

دراپ‌باکس یکی از قدیمی‌ترین و معروف‌ترین سرویس‌های فضای ابری است. استیو جابز پیش از راه‌اندازی iCloud سعی کرد دراپ‌باکس را خریداری کند که البته توافقی بین این دو حاصل نشد. رابط کاربری بسیار ساده، محیطی، تر و تمیز و کاربر پسند. امکان ذخیره سازی همه نوع فایل و به اشتراک‌گذاری آن‌ها، قابلیت همگام سازی فایل‌ها از طریق یک فولدر اختصاصی در سیستم عامل دسکتاپ شما و پشتیبانی از همه پلتفرم‌های جدید موبایل آن را به یکی از برترین سرویس‌های ارایه فضای ابری بدل می‌سازد. شما در ابتدا ۳ گیگابایت فضای ذخیره‌سازی رایگان در اختیار خواهید داشت که البته با دعوت از دوستانتان می‌توانید این فضا را به ۱۶ گیگابایت هم برسانید. مزیت این سرویس این است که بر خلاف وان‌درایو، گوگل درایو و آی‌کلاد شما را به یک اکوسیستم خاص سوق نمی‌دهد و مستقل از این که بیشتر از سرویس‌های کدام کمپانی استفاده می‌کنید می‌توانید از دراپ‌باکس استفاده کنید.

Cloud چگونه وارد دنیای تکنولوژی شد؟

نقش مهندس شبکه طراحی یک شبکه است که به درستی کار کند. برای این کار، زمان شخص، مختص به این خواهد بود که بررسی کند چه دستگاه‌های (Device هایی) در شبکه هستند و چطور به یکدیگر متصل شده‌اند تا آن‌ها را مدیریت و کنترل نماید. اما شبکه‌ها به شبکه‌های دیگر یا اینترنت هم متصل بودند. هنگامی که مهندسین شبکه، شبکه خود را بطور شماتیک طراحی می‌کردند برای نشان دادن این اتصال به عنوان بخشی از طراحی، سعی نکردند این شبکه‌ها را توصیف کنند! به این دلیل که بررسی‌های زیادی نسبت به این شبکه ناشناخته باید انجام می‌شد که این کار هم پیچیده بود هم زمان زیادی می‌برد و اصلا نیازی به این کار نبود! پس، این شبکه‌ها را با منحنی‌های بسته نشان دادند که رفته رفته تبدیل به شکل ابر یا همان Cloud شد و بیانگر چیزی بود که جزئیات آن ناشناخته است! به این صورت شکل ابر یا همان Cloud به دنیای تکنولوژی راه پیدا کرد!

درواقع Cloud استعاره‌ای از اینترنت شد و برای معرفی Cloud computing از عبارت "استفاده از داده‌ها از راه دور" استفاده گردید. اما اکنون اگر بخواهیم توضیح کامل و جامعی درباره این سوال که "Cloud computing" چیست؟ بدهیم باید گفت Cloud یک بستر برای سرویس دهی است! اما این که تنها بگوییم Cloud بستر سرویس‌دهی است یک دسته بندی کلی خواهد بود. پس، بخاطر وجود سرویس‌های مختلف، سرویس‌های Cloud به سه دسته بندی مهم تفکیک داده شده اند تا بتوان آن‌ها را راحت تر بررسی نمود.

میکرو سرویس ها

امروزه واژه **Microservice** بسیار مورد استفاده قرار می‌گیرد. مقالات و آموزش‌های زیادی پیرامون این موضوع تدوین شده و شرکت‌های بزرگی مانند Amazon بصورت بسیار گسترده از آن استفاده می‌کنند. اما ممکن است این مفهوم برای کسانی که تازه به دنیای برنامه نویسی وارد می‌شوند، کمی گنگ باشد. در این مقاله با این معماری و نقاط مثبت و منفی آن بیشتر آشنا می‌شویم.

Micro services

معماری نرم‌افزار یک راه‌حل علمی برای پیاده‌سازی یک سیستم نرم‌افزاری می‌باشد. به عبارت دیگر معماری نرم‌افزار مجموعه‌ای از تصمیمات است بر پایه‌ی عوامل موجود. این تصمیمات می‌تواند بر سرعت، امنیت و کنترل‌پذیری کل پروژه و درنهایت موفقیت آن تأثیر داشته باشد.

یک سیستم خرید آنلاین را در نظر بگیرید. این سیستم باید بتواند حجم وسیعی از درخواست‌های کلاینت‌های مختلف را در مدت زمان مناسب، پاسخ بدهد. یعنی افزایش تعداد کاربران فعال و یا حتی تعداد درخواست‌های آنان در روزهای خاص، نباید بر عملکرد سیستم خدشه وارد کرده و یا آن را از دسترس خارج کند. علاوه بر آن این برنامه‌ها باید بتوانند به سادگی رشد کرده و به‌روز شوند.

اکنون برای چنین سیستمی از چه معماری باید استفاده کنیم؟

برای ساخت و توسعه این نوع برنامه‌ها، دو مدل معماری رایج وجود دارد:

۱) معماری Monolithic
۲) معماری Micro services

معماری Monolithic

این نوع معماری تحت عنوان معماری MVC شناخته می‌شود.

- Model
- View
- Controller

برای روشن‌تر شدن این مسئله، مثالی می‌زنیم. فرض کنیم کاربری از طریق مرورگر خود درخواستی به برنامه بفرستد. این درخواست به دست سرور می‌رسد و لایه‌ی مرتبط با Controller این درخواست را گرفته و برای لایه‌ی Model ارسال می‌کند و این لایه هم با پایگاه داده ارتباط برقرار ساخته و داده‌های مرتبط را فراخوانی کرده و در صورت نیاز یک سری کارها بر روی داده‌ها انجام می‌دهد و در نهایت نتیجه را در اختیار Controller قرار می‌دهد. Controller پاسخ را در اختیار View قرار می‌دهد و مسئول نمایش پاسخ به کاربر (به وسیله ی مرورگر) است.

مشکلات معماری Monolithic

در معماری Monolithic یک هسته‌ی مرکزی داریم که کاربران با روش‌های مختلف می‌توانند با آن ارتباط برقرار سازند. اگرچه می‌توان کل سیستم را ماژول بندی کرد، اما همگی تحت یک پلتفرم واحد کنار یکدیگر قرار گرفته‌اند و امکان مجزاسازی این ماژول‌ها سخت و تا حدودی ناممکن است. در مورد استفاده از این معماری، باید مشکلات زیر را در نظر گرفت:

- با هر گونه تغییر در برنامه‌های مبتنی بر این معماری، نیاز به Build و Deploy مجدد کل برنامه می‌باشد که باعث از دسترس خارج شدن موقت برنامه خواهد شد.
- وجود باگ در یکی از ماژول‌ها، کل پروژه را تحت‌الشعاع خود قرار خواهد داد.
- امکان اجرای بخش‌هایی از سیستم بر روی کامپیوترهای دیگر نیست و اگر بخشی از سیستم از کار بیوفتد احتمال توقف عملکرد کل سیستم وجود دارد.
- کلیه اجزای سیستم بر اساس یک زبان برنامه‌نویسی مشخص داشته و برای یک فریم ورک مشخص نوشته می‌شوند. این برنامه‌ها اصطلاحاً چند سکویی نیستند و کامپونت‌های نوشته شده برای آن‌ها فقط در فریم ورک جاری قابل استفاده مجدد هستند.
- تنوع تکنولوژی (شامل زبان‌های برنامه‌نویسی، دیتابیس‌های مختلف، لایبری‌ها و فریم ورک‌های مختلف) وجود نخواهد داشت و اگر هم این‌گونه باشد، ارتباط برقرار کردن میان آنها دشوار خواهد بود.
- کامپونت‌ها را نمی‌توان به سادگی با یک معماری جدیدتر و بهینه‌تر جایگزین کرد؛ چون کل معماری را تحت الشعاع قرار می‌دهد.

البته باید در نظر داشت که روند توسعه‌ی اپلیکیشن‌های مبتنی بر این معماری، بسیار سریع‌تر از سایرین بوده و برای سیستم‌های کوچک، راه‌حلی مناسب و کارآمد می‌باشد.

معماری Micro Services

میکروسرویس روشی به منظور تقسیم بندی کردن یک برنامه به بخش‌ها یا سرویس‌های کوچک، سبک، مستقل است. به عبارت دیگر، میکروسرویس یک معماری توسعهی نرم‌افزار Distributed (پخش‌شده) است. این نوع سرویس‌ها صرفاً به منظور انجام دادن یک تسک خاص طراحی می‌شوند؛ به طور مثال، یک سرویس صرفاً وظیفه‌ی مدیریت کاربران را بر عهده دارد و سرویس دیگر برای کار با پایگاه داده استفاده می‌شود.

سرویس‌های مختلف یک اپلیکیشن با معماری میکروسرویسی با استفاده از درخواست‌هایی از جنس HTTP وAPI های به اصطلاح RESTfulاین ارتباط برقرار خواهد شد.

هر یک از این سرویس‌ها می‌توانند توسط تیم‌های جداگانه‌ای با پلتفرم توسعه و زبان برنامه‌نویسی و بانک اطلاعاتی جداگانه‌ای توسعه داده شوند و با یک مکانیزم سبک وزن مانند Http با یکدیگر در ارتباط باشند. این روش پیاده سازی قابلیت Scaling و Testability را بالا می‌برد و توسعه و نگهداری سیستم را آسان می‌کند.

مزایای استفاده از Micro Services

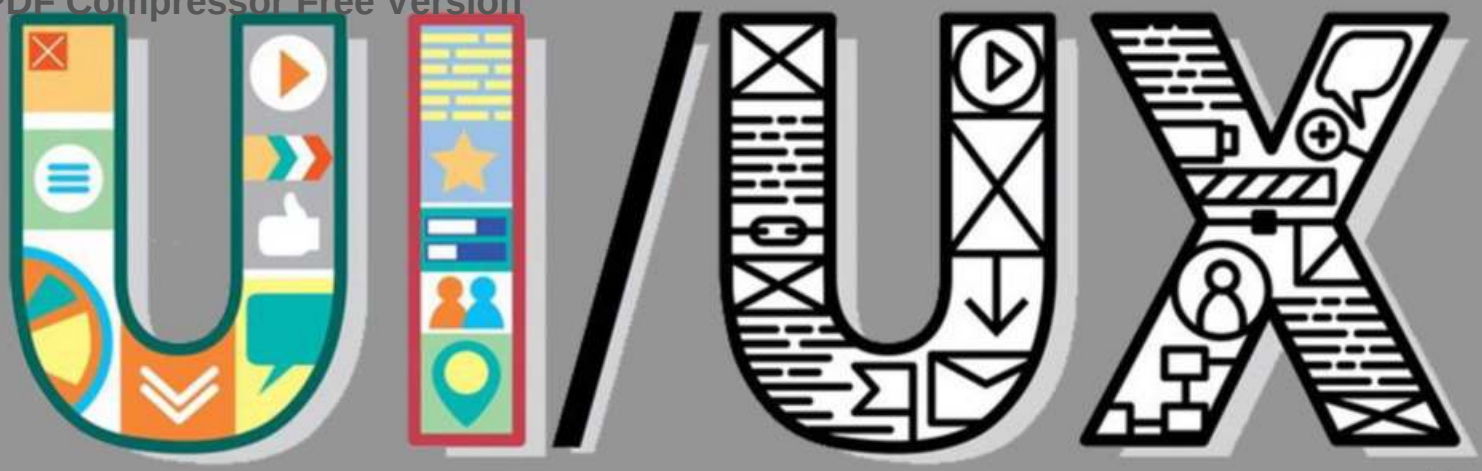
به طور کلی، ایده‌ی میکرو سرویس‌ها این است که این امکان به برنامه‌نویسان داده شود تا اپلیکیشن‌های خود را از اجزا یا سرویس‌هایی که مستقل از یکدیگر هستند و به سادگی قابل تغییر، حذف و به‌روزرسانی می‌باشند. ازمزایای دیگر این معماری می‌توان موارد زیر را نام برد:

- قابلیت Fault Isolation در سیستم افزایش می‌یابد.
- از آنجایی که سرویس‌ها از طریق زبان مشترک شبکه با یکدیگر در ارتباط هستند، می‌توان آنها را با زبان‌های برنامه‌نویسی مختلف و با فریم‌فرک‌های متفاوت نوشت.
- می‌توان هر سرویس را به صورت جداگانه ایجاد کرد و تغییر داد که باعث سرعت در به روزرسانی و فرآیند گسترش برنامه می‌شود.
- از آنجایی که این سرویس‌ها از طریق شبکه در تبادل هستند، می‌شود از آنها در سایر برنامه‌ها مجدداً استفاده کرد.
- ارتقاء و بروزرسانی هر یک از سرویس‌ها مستقل از کل سیستم بوده و به نسبت معماری monolithic زمان کمتری را صرف می‌نماید.
- درک عملکرد یک سرویس برای یک توسعه‌دهنده سیستم ساده‌تر از درک کل سیستم خواهد بود.
- مدیریت استفاده از منابع سخت‌افزاری در معماری میکروسرویس، بهینه‌تر است.

محدودیت های Micro Services

- به دلیل ارتباط سرویس‌ها در بستر شبکه، انتظار کندی عملکرد آن ها دور از ذهن نیست.
- به دلیل ارتباطات شبکه‌ای، احتمال آسیب‌پذیری امنیتی بالا می‌رود.
- به دلیل مجزا بودن بخش‌های مختلف برنامه، مدیریت و ردیابی عملکرد سرویس‌ها، یکی از کارهای اصلی به حساب می‌آید.
- بدلیل تعدد سرویس‌ها و بخش‌های مختلف، توسعه و تنظیمات آن سخت و زمان‌بر است.

در آخر باید اضافه کرد انتخاب نوع معماری بر عهده‌ی توسعه‌دهنده است. شما به عنوان یک برنامه‌نویس باید سیستم خود را به طور کلی در نظر بگیرید و با برآوردی هزینه و زمان و همچنین در نظر گرفتن ویژگی‌های برنامه‌تان، معماری مناسب را انتخاب کنید.



مقدمه:

همه ما مکالماتی را شنیده‌ایم که در آن از عبارات UIو UXاستفاده می‌شود. حال می‌خواهیم در مورد آن‌ها بطور مختصر توضیح دهیم که یعنی چی؟ تفاوت ui و ux چیست؟ و چه کاربردهایی دارند؟ طراحی ux یا همان طراحی User Experience به معنی طراحی تجربه کاربری است و طراحی ui یا همان طراحی User Interface به معنی طراحی رابط کاربری است. هر دوی این‌ها برای ارائه یک محصول کاملا ضروری هستند، هر دو در کنار هم! اما با این وجود قواعد مربوط به آن‌ها متفاوت است. طراحی تجربه کاربری، مبحثی تحلیلی و فنی‌تری است نسبت به رابط کاربری، که بیشتر مربوط به گرافیک، طرح و پاسخگویی طرح و چیزهای پیچیده‌تر است و اگر بخواهیم با یک مثال ساده توضیح دهیم، می‌گوییم :«اگر یک محصول دیجیتال را مانند بدن انسان فرض کنیم، استخوان‌بندی بدن همان رابط کاربری(کد ها) یا ui هستند که ساختار بدن را تشکیل می‌دهند و اندام‌ها همان تجربه کاربری یا ux هستند که عملکرد بدن را تشکیل می‌دهند.»

رابط کاربری یا UI چیست؟

در واقع UI عامل تعامل سیستم مورد نظر با کاربر است. عامل UIدر حالت‌های مختلفی بر روی انواع سیستم‌ها وجود دارد. از سایت گرفته تا نرم‌افزار، سیستم عامل و ... همگی به نوعی دارای رابط کاربری هستند. یک رابط کاربری می‌تواند مجموعه‌ای از دستورها یا تصویرها باشد. رابط کاربری یکی از مهم‌ترین بخش‌های یک برنامه می‌باشد، چرا که مشخص‌کننده‌ی سهولت یا دشواری یک برنامه جهت انجام پروسه مورد نیاز کاربر است. نوع بصری رابط کاربری که با عنوان GUIs شناخته می‌شود و مخفف عبارت Graphical User Interface است، متداول‌ترین نوع رابط کاربری می‌باشد که تعامل با برنامه را بسیار آسان و جذاب می‌سازد. تمرکز طراحان رابط کاربری بیشتر بر پیش‌بینی این موارد است که کاربر چه نیازهایی ممکن است داشته باشد و تلاش می‌کنند تا اطمینان حاصل کنند. رابط کاربری دارای عناصری است که دسترسی و فهمیدن آن‌ها برای کاربر آسان است. بهبود رابط کاربری یکی از اساسی ترین مباحث در دنیای IT می‌باشد به طور کلی UI شامل مفاهیمی چون طراحی تعاملی، طراحی بصری و معماری اطلاعات می‌باشد.

تجربه کاربری یا UX چیست؟

تجربه‌ی کاربری به اختصار UX شامل رفتارها، نگرش‌ها و احساسات یک کاربر درباره استفاده از یک محصول، سامانه یا سرویس خاص است. تجربه‌ی کاربری جنبه‌های کاربردی، تجربه شده، اثرگذار به صورت عاطفی، معنادار و ارزش‌مند و تعامل انسان و رایانه و مالکیت محصول را در برمی‌گیرد. علاوه بر این، برداشت یک فرد از جنبه‌های مختلف یک سامانه مانند سودمندی، کاربری آسان و کارایی نیز در گستره‌ی تجربه‌ی کاربری قرار می‌گیرد، به عبارت دیگر؛ تجربه‌ی کاربری همان خاطره‌ای است که با استفاده از یک محصول، سامانه یا سرویس در ذهن کاربر نقش می‌بندد طراحی تجربه کاربر شامل تعامل مرسوم انسان با کامپیوتر (HCI) بوده و این تعامل را با در نظر گرفتن تمام جنبه‌های یک محصول یا خدمات ارائه شده به کاربران گسترش می‌دهد.

چگونه یک رابط کاربری خوب طراحی کنیم؟

۱) طراحی به روش زیبایی‌شناسی:

یکی از راه‌های طراحی رابط کاربری این است که حس زیبایی‌شناسی کاربر را هدف قرار دهید؛ یعنی، ببینید چه چیزی باعث می‌شود که بخشی از وب سایت شما از دید کاربر زیبا باشد؟

انتون نیکو(Anton Nikolov) یک طراح UX این گونه می‌گوید: (پدیده‌ای وجود دارد که روانشناسان به آن می‌گویند: «تاثیر هاله‌ای». این بدین معناست که انسان‌ها با دیدن انسان‌هایی با ظاهر خوب، تصور می‌کنند که آن شخص اخلاقیات خوب و مثبتی هم در کنار آن دارد. همین مسأله در طراحی محصول نیز وجود دارد. محصولاتی که ظاهر بهتری دارند و از رابط کاربری بهتری برخوردارند، از نظر مشتری نیز ارزش و کیفیت بهتری دارند).

۲) طراحی به روش مینیمالیستی:

مینیمالیسم (به معنای به کم قانع بودن) مفهومی است که در رشته‌های زیادی از جمله هنر، موسیقی، ادبیات، طراحی و فلسفه دیده می‌شود. این نوع طراحی، شیک بودن را در سادگی خود دارد و چیزی که مهمتر است را در طراحی برجسته می‌کند. برخی از موضوعات اصلی در طراحی مینیمالیستی عبارتند از:

- **طراحی تخت یا Flat Design:** سبکی خاص از طراحی است که در آن تمامی جزئیات گرافیکی نظیر دکمه‌ها و آیکون‌ها، از نوع دو بعدی کار شده‌اند.
- **طراحی تک‌رنگ (Monochrome) یا با جعبه رنگی محدود:** این طراحی کمک می‌کند که تمرکز کاربر را برروی محتوای اصلی نگه‌داریم.
- **طراحی تایپوگرافی (Typography):** در این نوع طراحی، رنگ‌ها و فونت‌ها معمولا به صورت «Bold» و کاملا مشخص هستند.
- **محدودیت انتخاب (Choice Limitation):** در این نوع طراحی، با حذف عناصر و توابع غیر ضروری، توجه کاربر را به موضوع اصلی جلب می‌کنیم.
- **فضای منفی (Negative space) یا فضای سفید (whitespace):**این طراحی یکی از اصلی‌ترین طراحی‌های مینیمالیستی است. تضادی در این طراحی‌ها برقرار است که باعث می‌شود مطالب مهم سایت بیشتر در دید باشند.

۳) طراحی به روش فنی:

یک طراح به هر روش یا هر سبکی که کار کند، هدف نهایی او این است که یک رابط کاربری ارائه بدهد تا مردم بتوانند با رایانه‌ها ارتباط برقرار کنند. کری ودهاوس (Carey Wodehouse)در این مورد می‌گوید: قبل از لپ‌تاپ‌ها، تلفن‌های هوشمند و اپلیکیشن‌های همراه؛ قبل از رابط‌های کاربری گرافیکی، مرورگرها و موتورهای جست‌وجو؛ قبل از رابط کاربری و تجربه کاربری، چیزی داشتیم به نام «تعامل انسان با رایانه»، دانشی برای ارتباط بهتر و بصری بین انسان و رایانه.

اشتباهات رایج در طراحی رابط کاربری:

۱) کنتراست پایین در طراحی
۲) واکنش گرا نبودن طراحی
۳) کپی‌برداری
۴) معماری اطلاعات (Information Architect) نامناسب
۵) ناهماهنگی در سبک طراحی
۶) فرم‌های سنگین و کند

چند روش برای طراحی یک تجربه کاربری خوب:

۱) تحقیق و پژوهش(Research)
۲) ساخت شخصیت کاربر(User Persona)
۳) ساخت مسیر کاربر
۴) ساخت وایرفریم و پروتوتایپ
۵) طراحی رابط کاربری و تحویل به توسعه دهنده
۶) تحلیل و بررسی – آزمون و خطا

چگونه رفتار کاربران را برای ایجاد یک تجربه کاربری موفق درک کنیم؟

۱) تجزیه و تحلیل

۲) پشتیبانی مشتری

۳) آزمایش کاربردپذیری

جاوا ...

جاوا چی چی؟

یه زبان برنامه نویسیه؟

شاید یه شاخه از زبان جاواست!!!

سلام دوستان خوبم امروز می‌خواهم javascript رو به شما معرفی کنم.

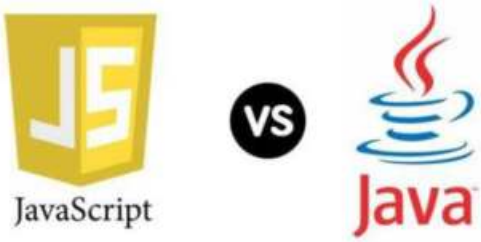
جاوا اسکریپت چیست؟؟؟

جاوا اسکریپت زبانی برای نوشتن برنامه‌های ساده‌ایست که در مرورگر کاربر اجرا می‌شوند و ظاهر و امکانات وب‌سایت‌ها را متحول می‌کنند. جاوا اسکریپت تقریباً در هر وب‌سایتی کاربرد دارد و در عین حال کمتر نامش را شنیده‌ایم! سال‌ها پیش در دورانی که صفحات وب به صورت HTML و بسیار ساده طراحی می‌شدند، متن و عکس در کنار هم قرار می‌گرفت و حالت پویا نداشت، در واقع همه چیز حالت ایستا داشت. اما این روزها، این موضوع برعکس شده و کمتر سایتی را سراغ داریم که به طور کامل ایستا باشد، ولی وقتی صفحه‌ی شبکه‌های اجتماعی مثل فیس‌بوک و توییتر را باز می‌کنید، جاوا اسکریپت روی مرورگر شما اجرا می‌شود. به همین علت است که وقتی با استفاده از موس سراغ بخش‌های مختلف صفحه می‌روید، برخی عناصر عوض می‌شوند.

جاوا اسکریپت شاخه‌ای از جاواست؟

خیر؛ ممکن است فکر کنید جاوا که در دوران گوشی‌های غیرهوشمند در اکثر گوشی‌ها متداول بود، همان جاوا اسکریپت است؛ اما این تصور کاملاً اشتباه است. جاوا زبانی است که توسط کمپانی Sun Microsystems توسعه داده شده است.

جاوا اسکریپت به شما امکان ساخت سایت‌های تعاملی را می‌دهد؛ پس کاملا با جاوا متفاوت است. جاوا اسکریپت در کنار HTML و CSS به یک تکنولوژی ضروری وب تبدیل شده است و اکثر مرورگرها جاوا اسکریپت را اجرا می‌کنند؛ بنابراین، اگر می‌خواهید یک توسعه‌دهنده‌ی وب باشید نیاز است تا جاوا اسکریپت را بیاموزید. همچنین اگر می‌خواهید یک توسعه‌دهنده‌ی frontend باشید و یا از جاوا اسکریپت در توسعه‌های backend استفاده کنید باید بر **جاوا اسکریپت تسلط داشته باشید. علاوه بر این موارد؛** امروزه جاوا اسکریپت به حدی گسترش یافته که در توسعه‌ی برنامه‌های موبایل، ویندوز و حتی بازی‌ها از آن استفاده می‌شود. در کل یک زبان بسیار محبوب و یادگیری آن یک مهارت بسیار کاربردی است.



تاریخچه جاوا اسکریپت

در سال ۱۹۹۵ دو مرورگر اصلی در دنیای وب وجود داشت. مرورگر اکسپلورر که تازه کارش را شروع کرده بود و مرورگر دیگری به نام Netscape کمپانی نت‌اسکیپ علاوه بر معرفی یک مرورگر کامل و محبوب، به دنبال ایجاد یک زبان برنامه‌نویسی هم بود؛ که ساده باشد و برای توسعه‌دهندگان و حرفه‌ای‌ها، مفید واقع شود و در ادامه دنیای وب و وب‌سایت را متحول کند.

جاوا اسکریپت زبانی بود که پس از ۱۰ سال کار Brendan Eich متولد شد، کسی که بعدها برای مدتی کوتاه به عنوان رهبر موزیلا فعالیت کرد. زبانی که او ابداع کرده بود برای انواع برنامه‌نویسی شیءگرایی آبجکتیو و تابعی کاربرد داشت و از طرفی به زبان‌های متداول روز مثل C و C++ شبیه است.

آینده جاوا اسکریپت

مرورگر وب یکی از پراستفاده‌ترین برنامه‌ها هم روی دسکتاپ و هم روی دستگاه‌های موبایل است. با افزایش سرویس‌دهی‌ها در وب، به محبوبیت جاوا اسکریپت افزوده و به‌طور کلی باعث ابداعات تکنیکی زیادی در جاوا اسکریپت می‌شود. بسیاری از توسعه‌دهندگان و تاجران از جاوا اسکریپت برای گسترش محصولات وب خود در خارج از مرورگر استفاده کرده‌اند. این به این معناست که از جاوا اسکریپت می‌توان در تولید نرم‌افزارهای دسکتاپ و موبایل استفاده کرد.

جاوا اسکریپت هشتمین زبان پرطرفدار دنیاست در شاخص TIOBE.

توانایی یک زبان برای باقی ماندن و زنده ماندن بستگی به این دارد که آیا به زبان ایده‌های نو و خلاقانه وارد می‌شود یا نه.

علاقه به جاوا اسکریپت در سال ۲۰۱۵، ۲۲,۳٪ افزایش یافت.

با توجه به این‌که سرویس‌های آنلاین در حال افزایش است؛ واضح است که جاوا اسکریپت به رشد خود ادامه خواهد داد و در سال های آینده هم همچنان باقی خواهد ماند.

در کل پیش‌بینی این‌که آیا مسیر جاوا اسکریپت به عنوان یک تکنولوژی همیشه در حال گسترش باشد یا نه دشوار است، اما نیاز به گفتن نیست که جاوا اسکریپت تا زمانی که تقاضا برای آن وجود داشته باشد پیشرفت خواهد کرد.

و در آخر فرصت شغلی در جاوا اسکریپت

با توسعه‌ی روزافزون کسب و کارهای آنلاین، تقاضا برای توسعه‌دهندگان جاوا اسکریپت نیز در حال افزایش است. همچنین جاوا اسکریپت پرتقاضاترین مهارت در Angel List است.

اگر می‌خواهید از جاوا اسکریپت برای توسعه‌های backend استفاده کنید، متوسط حقوق سالانه در ایالات‌متحده برای توسعه‌دهندگان Node.js حدود ۹۸،۹۶۲ دلار است.

مصاحبه با

موفق

مصاحبه‌ی این شمارمون با سرکار خانم مریم پورعبادی، یکی از فارغ‌التحصیلان موفق دانشگاهمون هست، ایشون دوره‌ی کارشناسی رو در دانشگاه الزهرا گذروندن در حال حاضر محقق رشته‌ی کامپیوتر در دانشگاه کالیفرنیا، سندیگو هستن. ایشون در طول تحصیلشون زمینه بسیاری از فعالیت‌های فرهنگی، علمی و آموزشی رو برای گروهمون فراهم کردن. صحبت‌های ایشون رو در این شماره می‌خونید.

- **لطفاً یه بیوگرافی کلی از خودتون برامون بگید.**

مریم پورعبادی هستم. آذر ۱۳۷۰ در تهران متولد شدم. فرزند اول یک خانواده‌ی فرهنگی‌ام و یک برادر به اسم مهدی و یک خواهر به اسم معصومه دارم. میرزاقاسمی غذای مورد علاقه منه و تالش سرزمین محبوبمه.

یادم می‌آد نوروز سه سال پیش یکی از بستگانم بهم گفت «چهارم دبستان بودی که دیدمت پای کامپیوتر نشستی و سعی می‌کنی از ساده‌ترین برنامه‌ی ویندوز، پیچیده‌ترین خروجی‌های ممکن رو بگیری. اونجا بود که مطمئن شدم تو حتماً یه روز لیسانس مهندسی‌کامپیوتر می‌گیری!»

من اما فکر می‌کنم علاقه‌ام به رشته‌ی کامپیوتر به صورت جدی از سال دوم راهنمایی با شرکت توی المپیاد ریاضی و کامپیوتر و آوردن رتبه شروع شد و اونجا بود که فهمیدم نه فقط از روی کنجکاوی زمانم رو صرف یاد گرفتن کوچک‌ترین جزییات و یا پیچیده‌ترین ترکیب‌ها می‌کنم، بلکه استعداد تحلیل و یادگیری مفاهیم پایه‌ای این مبحث و همین‌طور توانایی پیاده‌سازی ایده‌هام به کمک این مفاهیم رو هم دارم. با شروع دوره‌ی دبیرستان در مدرسه‌ی استعدادهای درخشان فرزانگان تهران، بیشتر درگیر برنامه‌نویسی شدم و بعد هم با شرکت توی مسابقات رباتیک فوتبالیست دوبعدی و سه‌بعدی، انگیزه‌ام برای ادامه دادن این شاخه دوچندان شد.

با توجه به علاقه‌مندی خاصی که به رشته کامپیوتر داشتم، تحصیلات دانشگاهیم را در این رشته شروع کردم. در سال ۱۳۸۹ در دانشگاه الزهرا (س) در رشته مهندسی کامپیوتر پذیرفته شدم و در سال ۱۳۹۳ فارغ‌التحصیل شدم و بعد برای ادامه تحصیل به آمریکا رفتم و دوره دوساله‌ی کارشناسی‌ارشدم رو در رشته‌ی علوم‌کامپیوتر شروع کردم. بعد از فارغ‌التحصیلی از این مقطع، در سال ۱۳۹۶ تحصیلاتم رو در مقطع دکترای علوم و مهندسی‌کامپیوتر در دانشگاه University of California, San Diego شروع کردم و در حال حاضر مشغول به مطالعه و ساخت ربات‌های اجتماعی برای محیط‌های پزشکی در یک گروه تحقیقاتی هستم. ساختن نسل جدید ربات شبه‌بیمار که قابلیت انسان‌نمایی، ابراز احساسات، درد و علائم بیماری رو در ظاهرشون داشته باشن پروژه‌ایه که طی یک سال اخیر دارم روش کار می‌کنم.

- **از دانشگاه الزهرا برامون بگید. اینکه چجوری اومدید به این دانشگاه، چی شد که رشته‌ی کامپیوتر رو انتخاب کردید؟ درباره رشته و دانشگاه‌تون چه حسی داشتید؟**

همون‌طور که قبلاً هم گفتم، از اول علاقه خاصی به رشته کامپیوتر داشتم تا جایی که حاضرنبودم با رشته‌هایی مثل پزشکی عوضش کنم! البته الان خیلی علوم به هم مرتبط شدن و می‌شه لذت چندتاشون رو با هم برد! مثلاً امروز به عنوان یه محقق کامپیوتری داشتم فرایندی به اسم آزمایش داوینچی رو روی یک ربات جراحی تست می‌کردم و با حرکت دست‌هام و همین‌طور به کمک الگوریتم‌های افزایش سرعت و واقعیت مجازی، عملیات جراحی رو کنترل می‌کردم. جالب‌تر این‌که مسئول پروژه بهم گفت:((من فکر می‌کنم که اگر تو توی فیلد پزشکی هم می‌رفتی، قطعاً یک جراح خبره می‌شدی!)) حرفش خیلی بهم چسبید! اعتقادم اینه که هرکس در هر رشته‌ای و هر دانشگاهی اگه بر اساس علاقه‌مندی رشته‌اش را انتخاب کنه، حتماً توش موفق می‌شه و می‌تونه به بهبود کار و شرایط گروه‌های دیگه‌ی شغلی هم کمک کنه.

من چهار سال از عمرم را صرف تحصیل در رشته‌ی مهندسی توی دانشگاه الزهرا (س) کردم و تعلق خاطر خاصی نسبت بهش دارم. دانشگاه الزهرا (س)، دانشگاه متفاوتیه و این تفاوت توی زندگی من تاثیر مثبتی داشته. از اساتید بزرگوارم خیلی چیزها آموختم و قطعاً بخش اعظمی از موفقیت‌هام رو مرهون راهنمایی‌های اون بزرگواران می‌دونم.

به علاوه دانشجویان این دانشگاه از سطح دانش بسیار بالایی برخوردارن و در زمان تحصیل من اکثر دانشجوها با رتبه‌های بسیار خوبی وارد این دانشگاه می‌شدن به نحوی که امکان پذیرش برای رشته‌هایی مثل علوم‌کامپیوتر و یا ریاضی در دانشگاه‌های شریف و یا تهران رو داشتن و اکثراً بنا به معیارهای مطلوب‌شون مهندسی رو انتخاب کردن و به این دانشگاه اومدن.

- **از دوران تحصیل‌تون در دانشگاه برامون بگید.**

یکی از ویژگی‌هام که خیلی دوسش دارم اینه که همیشه سر همه‌ی کلاس‌ها حاضر می‌شدم و مطالب رو همون‌جا توی کلاس درس یاد می‌گرفتم، در نتیجه عمده‌ی وقت درسی خارج از کلاسم، فقط صرف انجام تکالیف می‌شد. همین هم باعث می‌شد با حداقل زمانی که برای درس خوندن در کنار بقیه‌ی فعالیت‌های علمی، فرهنگی و تفریحی‌ام

صرف می‌کردم، بازم بتونم در پایان دوره‌ی کارشناسی با معدل الف فارغ‌التحصیل بشم. این روند طی دوره‌ی کارشناسی‌ارشدم هم ادامه پیدا کرد و به عنوان دانشجوی ممتاز با رتبه‌ی اول فارغ‌التحصیل شدم و به دوره‌ی دکترا راه پیدا کردم.

در کنار درس خوندن، درس دادن هم یکی از بزرگ‌ترین لذت‌های زندگی منه و به خاطر همین تا به امروز، TA بیش از ۱۵ درس بودم. بخت با من یار بود که بعد از نشستن سر چندین کلاس درس دکتر بهروز قلی‌زاده تونستم به سطحی از دانش برسم که از اون به بعد برای دو تا از درس‌های ایشون هم به عنوان حل تمرین انتخاب بشم و به دانشجوهاشون کمک کنم.

- **شما در دوران دانشجویی‌تون خیلی فعال بودید... چجوری خودتون و زمان‌تون رو مدیریت می‌کردید؟ فعال بودن در زمینه‌های غیر از درس در دانشگاه به نظرتون خوبه؟**

زندگی ما آدم‌ها خواه‌ناخواه از ترکیب اجزای مختلف به هم تشکیل شده. مثل تحصیل، کسب درآمد، روابط خانوادگی، روابط اجتماعی، تفریح و خیلی چیزای دیگه. به خاطر وجود همین اجزای مختلف زندگی آدم‌ها (مستقل از این‌که کسی بخواد در کنار درس، فعالیت غیر درسی داشته باشه یا نه)، برای هر آدمی لازمه که مدیریت کردن شرایط، زمان و امکانات رو بلد باشه. مدیریت یک مهارته که بر اساس یادگیری و تکرار و تمرین به مرور زمان در افراد ایجاد و تقویت می‌شه.

یکی از بزرگ‌ترین نعمات زندگی من، پدر و مادرم هستن که هر دو از اساتید و مدیران موفق کشورند. به لطف داشتن‌شون و روش تربیتی‌شون، من در طی سال‌ها یاد گرفتم که باید چندین مهارت اصلی رو یاد بگیرم و متوقف نشم تا زمانی که به چیزی که می‌خوام دست پیدا کنم. مهم‌ترین نکاتی که می‌تونم به همه‌ی شما هم توصیه کنم که تمرین‌شون کنین اینان:

۱) داشتن هدف و برنامه:

من یاد گرفتم اهداف نهایی‌ام رو از بین هدف‌هایی که برای من با ارزشن انتخاب کنم تا در آخر کار احساس کنم برآیند هزینه‌هایی که در این راه کردم با دستاوردهایی که داشتم مثبت بوده و در مجموع چیزی به دارایی‌های من (اعم از سواد، شادی، سلامت، ثروت، اعتبار اجتماعی و یا احساس رضایت) اضافه شده.

برنامه‌ریزی‌های کوتاه‌مدت و بلندمدت و به‌روزکردن‌شون بر اساس تغییرات در طی زمان به دو دلیل اصلی مهمن: اول به این خاطر که یک سری اقدامات پیش‌نیاز اقدامات دیگه هستند؛ و دیگه به خاطر این‌که تفاوت‌ها و یا تداخلی ممکنه بین برنامه‌ریزی‌های اشخاص مختلف درون گروه و یا سایر گروه‌ها وجود داشته باشه که باید قبل از شروع، یک مرحله شناسایی بشن. این برنامه‌ریزی کمک شایانی به ذخیره کردن زمان و استفاده‌ی بهینه از اون در آینده می‌کنه.

برای این کار لازمه مشخص کنیم چه مراحلی برای رسیدن به یک هدف، اساسی هستن و در هر مرحله چه ویژگی شخصیت‌مون بیشترین کمک رو می‌تونه به پشت سر گذاشتن اون مرحله بکنه. پیش‌نیازش هم اینه که درباره‌ی خودمون، توانایی‌هامون و استعدادهامون به شناخت برسیم و ببینیم در چه قسمت‌های کار و زندگی می‌تونیم موثرترین باشیم.

۲) اولویت‌بندی کارها

می‌گن ایده‌های بد باعث از بین رفتن یک کسب و کار نمی‌شه، بلکه شروع کردن میلیون‌ها ایده‌ی خوب به صورت همزمان و تموم نکردن‌شون باعث شکست می‌شه.

کافیه کسی درباره یه موضوع شروع به حرف زدن با شما بکنه، بلافاصله حجم زیادی از فکرها و ایده‌ها به سمت مغز خوب‌تون هجوم می‌آرن! این درحالیه‌که یه نفر نمی‌تونه همه‌ی کارها رو همزمان به نحو احسنت انجام بده و الزاماً همه‌ی ایده‌هاش شدنی یا درست نیستن.

برای همین من یاد گرفتم کارهام رو اولویت‌بندی کنم و زمان محدودم را صرف کارهایی که برام ارزش بیشتری دارن، بکنم. یه دفترچه دارم که همیشه همراهمه، توش برنامه‌هام رو می‌نویسم، بسته به اهمیت‌شون بهشون شماره می‌دم و بعضاً وقتی ددلاین هدف‌های اصلیم نزدیک می‌شه، مراحل باقی مونده رو روی استیکرها می‌نویسم و جاهایی که بیشتر جلوی چشمم می‌چسبونم تا زمانی که به اون هدف برسم. همین فرایند رو برای هدف‌های مربوط به گروه انجمن علمی دانشجویی کامپیوتر هم پی گرفتیم که فکر می‌کنم همین هم یکی از دلایل اصلی نتیجه گرفتن و شکوفایی این گروه شد.

۳) داشتن انگیزه

روزهایی هست که ممکنه سرما خورده باشی، خسته باشی، از پروژه‌ات نتیجه‌ای که باید رو نگرفتی و یا حتی بی‌دلیل بی‌انگیزه باشی و ترجیح بدی برای یک مدت هیچ کاری نکنی. توی همچین شرایطی افراد اطرافت احتمالاً نمی‌تونن کار زیادی برات بکنن.

من یاد گرفتم چیزی که «مریم» رو از سایرین در شرایط مشابه متفاوت می‌کنه، اینه که مریم خودش، به خودش انگیزه می‌ده تا کاری برای تغییر شرایط بکنه. این یادگیری توی دنیای حرفه‌ایم هم خیلی برام کاربردی شده، چون اگر توی فرازونشیب های درس و یا کار قرار بگیرم، خودم رو به طور کامل مسئول نتایج کار خودم می‌دونم، در نتیجه به روش‌های مناسب به خودم انگیزه می‌دم که اصولاً باعث شادی درونم هم می‌شه!:-) و این ناخودآگاه باعث می‌شه راه‌های جدیدی برای حل مسئله‌ی پیش روم پیدا کنم.

یه موردش که الان یادم می‌آد مربوط به زمانی می‌شه که به عنوان مدیر مسئول نشریه‌ی پردازش و با همکاری شبانه‌روزی جمعی از

دانشجوها، اولین شماره‌ی نشریه‌ی پردازش رو به منظور گسترش دانش روز مرتبط به علوم کامپیوتر چاپ کرده بودیم، ولی برخلاف انتظارمون استقبال چندانی در زمان فروش نمی‌شد. خاطرم هست که طی فروش دو روز اول نشریه شماره سوم سال ۱۳۹۲، بچه‌های تیم هر روز در ساعت‌های مشخص شده توی غرفه‌ای نزدیک به سلف دانشگاه حاضر می‌شدن، اما به خاطر استقبال کم دانشجوها، تیم هم داشت انگیزه‌اش کمرنگ می‌شد.

برای همین صبح روز سوم یک تعداد هندوانه‌ی سرخ تهیه کردم و خودم به تیم فروش ملحق شدم. تبلیغ اون روز تبدیل شد به «نشریه به شرط چاقو» و هربار به هرکسی که نشریه رو می‌خرید و همین‌طور به اعضای تیم یه قاچ هندونه تعلق می‌گرفت!:-) با همین ایده‌ی ساده نه تنها تمام نسخه‌های اون شماره به فروش رفت، بلکه توی اون هوای گرم، خنکی اون هندوانه‌ی شیرین برای تیم «پردازش» خاطره‌ای موندگار و دلیل انگیزشی برای بهبود سطح رضایت‌شون از بیشتر دیده شدن ماحصل کارشون توسط دیگران شد.

۴) تمرکز

یک شبانه روز برای همه‌ی آدم‌های روی کره‌ی خاکی زمین، ۲۴ ساعته. پس این به نظر یک بهانه می‌آد که برای انجام یک کار بگیم: وقت کم آوردیم، درحالی‌که آدم‌های دیگه‌ای هستند که تونستن همون کار یا مشابه اون رو انجام بدن. عاملی که باعث تفاوت در میزان کارآمدی ۲۴ ساعت آدم‌های مختلف می‌شه، اینه که آیا بر روی جهت درستی که مد نظرشونه، تمرکز می‌کنند یا نه. طی سال‌های کارشناسی‌ام سعی کردم که تمرکز کردن رو به عنوان یکی از هنرهای مربوط به مدیریت زمان تمرین کنم.

توی چهار سال، مشغول کارهای متفاوت زیادی شدم، از جمله عضویت در گروه درسی پروژه‌ی شش ماهه‌ی تولید نرم‌افزار، حل تمرین دروس متعدد، مدیر مسئول و طراح گرافیک نشریه، مدیر انجمن علمی دانشجویی کامپیوتر، تورلیدر گروه‌های تفریحی و مسئول برگزاری دوره‌های آموزشی. برای همین برای انجام کارهای مختلف، زمان‌های جدایی در طی روز و هفته اختصاص می‌دادم و به امور و وظایف مربوط به هر کار به صورت جداگانه می‌پرداختم که فکر می‌کنم همین امر هم به پیشرفت کارهای گروهیم کمک شایانی کرد. می‌خوام بگم اگرچه می‌گن خانوم‌ها می‌تونن چندین کار رو به صورت هم‌زمان انجام بدن، اما این به این معنا نیست که الزاماً همه‌ی آن‌ها را به همان خوبی انجام خواهند داد که جدا جدا برای هر کدوم وقت بگذارن!

۵) قدرت تصمیم‌گیری

اینکه تصمیم بگیری با چه کسی و در چه زمانی ملاقات بکنی، چه نیروهایی رو برای انجام چه فعالیت‌هایی جذب کنی، چه قدر زمان و منابع انسانی و مالی و سازمانی داری و چقدرش رو می‌تونی برای چه کاری اختصاص بدی، چه کاری رو به پایان برسونی و چه کاری رو رها کنی، همه‌ی

این‌ها تصمیم‌های سختی هستن که گرفتنشون در زمان مناسب مهمه، چرا که نه فقط روی زندگی خودمون، بلکه روی حال و آینده‌ی سایرین که باهامون در تعامل و همکاری هستند هم تاثیر می‌ذاره. برای من سخت‌ترین بخش در بین کارهایی که توی اون سال‌ها انجام دادم همین بود که سعی کنم تصمیماتی که می‌گیرم، منافع جمعی افرادی که باهاشون در تعاملم اعم از حدود ۱۵ نفر عضو تحریریه‌ی نشریه، ۲۰ نفر فعال در انجمن، ده‌ها دانش‌آموز کلاس‌هایی که حل تمرینشون بودم و تمام دانشجوهای فنی و مهندسی دانشگاه و به صورت خاص گروه کامپیوتر رو تامین کنه.

۶) تعامل اجتماعی با دیگران

برای موجودی اجتماعی به نام انسان، تقویت مهارت برقراری ارتباط و مذاکره کردن، لازمه‌ی پیشرفت توی زندگی اجتماعیشه. شما برای شناخته شدن خودتون و کارتون و همین‌طور گرفتن حمایت‌های مالی و انسانی و حتی عاطفی، نیاز به شناخت و تعامل با افراد و گروه‌های مختلف در سطوح مختلف (از جمله رییس دانشگاه، مدیر همایش‌های ملی کشوری و مسؤل حراست یا برنده‌های نوبل) دارین. برای همین خوبه که همیشه گوشه‌ی ذهنمون داشته باشیم که اقدامات و لینک‌های به ظاهر کوچک، یک زمانی می‌تونن نتایج خوب و قابل توجهی رو برامون به همراه داشته باشند. ملموس‌ترین مثالی که احتمالاً همه‌ی شما هم تجربه‌اش کردین، کمک‌های سربه‌زنگ آقای حاج‌آقایی هست، در شرایطی که شاید بعضاً جز شرح وظایفشون هم نبود، اما دریغ نمی‌کردند.:)

در مجموع فعال بودن در زمینه‌های غیر درسی در دانشگاه می‌تونه کمک بزرگی در راستای تمرین کردن برای بهبود این توانایی باشه و به علاوه کمک کنه تا افراد توانایی (نه) گفتن در شرایط لازم رو در خودشون تقویت کنن.

۷) مفیدتر بودن

اگر کسی می‌خواد خودش و کارش رو به بالاترین حد استانداردهای موجود برسونه، بایستی هر چیزی که از حد استاندارد اون شخص پایین‌تره به چالش بکشه. برای مثال اگه شخصی وظیفه‌ای رو به شما محول می‌کنه اما شما به هر دلیلی فکر می‌کنید اون وظیفه، کاری مناسب با توانمندی‌ها یا خواسته‌های شما نیست، بایستی از کسی که بهتون این کار رو محول کرده درباره‌ی چرایی این انتساب سوال بپرسین و یا دلایل خودتون رو برای عدم تمایل برای انجام اون کار ابراز کنید. این باعث می‌شه که بخش زیادی از زمان شما برای کارهایی که برای شما مفیدتره و یا شما تخصص بیشتری در انجامش دارین، ذخیره بشه.

متاسفانه این یکی از شایع‌ترین اتفاقاتی هست که می‌افته: در واقع در این حالت فرد بالادست، کاری رو به شما محول می‌کنه، فقط به خاطر این‌که این آسون‌ترین اقدام ممکنه برای اون شخصه؛ نه به خاطر این‌که این بهترین و سنجیده‌ترین تصمیم برای شماست. بزرگی می‌گه «زمان با ارزش‌ترین دارایی هر انسانه»، پس بهترین کار قبل این‌که بخواید زمانتون رو صرف کاری که بهتون واگذار شده بکنین، اینه که با توجه به عوامل مختلف ببینید، انجام این کار تا چه

حد شما رو به موفقیت نزدیک می‌کنه. (تعریف موفقیت به عهده‌ی خودتون!:-))

۸) مهار استرس

برای هرکسی در هر جایگاهی ممکنه شرایطی پیش بیاد که منطبق با پیش‌بینی‌هاش نبوده باشه و فرد دچار نگرانی و اضطراب بشه.

این اضطراب می‌تونه روی ذهن شما، عملکرد فعلی‌تون و تصمیم‌های بعدی که می‌گیرین اثر بذاره. همین‌طور این استرس توی ظاهر شما هم تاثیر خودش رو نشون می‌ده و این می‌تونه قضاوت ظاهری که دیگران درباره‌ی شما می‌کنن رو به سمت بی‌اعتمادی ببره. پس سعی کنیم استرس‌مون رو مدیریت کنیم!:-)

برای من به عنوان مدیر انجمن علمی دانشجویی کامپیوتر، لازم بود تا به امور علمی، فرهنگی، مالی، اداری و مدیریتی آشنا باشم و متناسب با شرایط مختلف تصمیمات مناسب بگیرم. گاهی یک تصمیمی که گرفته می‌شد، ممکن بود منطبق با علایق فعالان فرهنگی باشه، اما برای گروه مالی به نظر جالب توجه و اجرایی نیاد. چیزی که کمک کرد تا اکثراً بتونم رضایت و همکاری گروه‌های مختلف رو برای اجرایی شدن برنامه‌هایی که انجمن تصمیم به برگزاریش داشت بگیرم، این بود که؛ همیشه در ملاقات با این گروه‌ها سعی می‌کردم هم خودم و هم اعضای تیم با اعتمادبه‌نفس بالا ظاهر بشیم و برنامه رو همراه با هدف نهایی و مشکلات احتمالی که ممکن بود به همراه داشته باشه باهاشون صادقانه در میون بذاریم.

همین امر باعث می‌شد به خودمون، توانمندی‌هامون، برنامه‌هامون و به پیشنهاداتمون اعتماد کنن، منابعشون رو در اختیارمون بذارن و حمایتمون کنن تا به هدف‌های پیش‌بینی شده نزدیک‌تر بشیم.

۹) کار گروهی

من همیشه فردی اجتماعی بوده و هستم:-) و این مسئله همیشه دو پیامد فوق‌العاده رو برام به همراه داشته: پیشرفت و شادی بیشتر!:-)

تجربه‌ی من این رو می‌گه که اگر تیم مناسبی تشکیل داده باشید، کاری که شاید به تنهایی امکان

انجام تمام و کمالش نبوده، با همکاری در فعالیت گروهی، با کیفیت بالاتر و در زمان کمتری انجام می‌شه. جالب‌تر از اون اینه که اگر چه افراد در یک تیم همه مثل هم نیستند و فکرها، خواسته‌ها و نیازهای بسیار متفاوتی با هم دارن، اما همه‌ی اعضای تیم یک‌سری ویژگی‌های مشترک با هم دارند که همین مسئله اون‌ها رو در کنار هم جمع می‌کنه و باعث افزایش روحیه‌ی همکاری‌شون می‌شه.

مثل ویژگی تمایل به خوشحال کردن و خوشحال شدن، گرایش به هنر و زیبایی و یا علاقه به یاد گرفتن مطالب اضافه بر درس. مشارکت در کارهای داوطلبانه که مازاد شرح وظایف کاری هر کسیه، می‌تونه لذت اعضای تیم رو از محیط کاری و یا درسیشون افزایش بده و دل‌های آدم‌ها رو به هم دیگه نزدیک‌تر کنه.

یادمه یکی از کارهایی که خیلی مورد استقبال افراد تیم انجمن کامپیوتر و نشریه قرار گرفت، این بود که هر از چندگاهی دورهمی‌های زنگ تفریح طور داشتیم! اون زمان‌ها که انجمن تازه پا گرفته بود، اگر می‌خواستیم مسابقه‌ای برای بیشتر درگیر کردن دانشجوها با امور علمی و فرهنگی برگزار کنیم، منابع مالی کافی برای تهیه‌ی جایزه‌اش نداشتیم، برای همین توی این دورهمی‌ها به همدیگه روش‌های درست کردن دستبندهای مختلف رو یاد می‌دادیم و وقتی که این دستبندهای رنگی خیلی زیبا آماده می‌شد، اون ها رو هرماه به عنوان جایزه به برنده‌های مسابقات می‌دادیم. توی یکی دیگه از این دورهمی‌ها، یک روش برای بافتن شال‌گردن یاد گرفتیم و به همه‌ی اعضای تیم برگزارکننده‌ی برنامه‌ی «جشن روز مهندسی» که تازه بناش رو شروع کرده بودیم دادیم تا خوش‌رنگ‌تر و متحدتر بشن:-)

۱۰) تهیه‌ی گزارش

یکی دیگه از عواملی که به من کمک می‌کرد تا بتونم ذهنم و کارهام رو مرتب کنم، آماده و به‌روزکردن گزارش از تقریباً همه چیز بود! از رکورد کردن کتاب‌ها و صندوق مالی برای سیستم «جابه‌جایی کتاب‌های علمی بین دانشجوها و فارغ‌التحصیل‌ها» که طی سال آخر کارشناسیم راه اندازیش کردم، تا جمع کردن و آرشیو کردن تمام فرم‌های اداری که پر کرده بودیم و شرح وظایف افراد در هر تیم که توی دفتر ثبت شده بود.

داشتن یک سیستم برای نظم دادن به پرونده‌ها، ذخیره‌ی اطلاعات و گزارش‌ها، نه تنها باعث بهبود عملکرد خودمون و گروه‌مون در زمان حال می‌شه، بلکه افراد بعد از ما هم می‌تونن از این منابع استفاده کنن و کارشون رو بر پایه‌ی کارهایی که ما انجام دادیم بنا کنن و پیش ببرن. به این ترتیب با استفاده از این روش، زمان قابل توجهی رو برای آیندگان ذخیره می‌کنیم که در صورت عدم وجود این رکوردها می‌بایست مجدداً این راه‌ها رو آزمون و خطا طی کنن.

۱۱) صبر و بردباری

تمرین کنیم که صبور باشیم:-) مهارت مدیریت به معنی انجام دادن همه‌ی کارها نیست، بلکه به معنی به پایان رسوندن کارهای مهمه.

عجله کردن احتمال بروز اشتباه در کار رو افزایش می‌ده و معمولاً جبران کردن این اشتباهات پیش اومده در نهایت منجر به صرف زمان بیشتری می‌شه در مقایسه با زمانی که همان کار با صبر و حوصله بیشتر انجام می‌شد.

۱۲) بخشش

بخشش چیزی نیست که ناخودآگاه خودش پیش بیاد، ما آدم‌ها باید انتخاب کنیم که ببخشیم.

من نیاز به این مهارت رو این روزها که در پروژه‌های بزرگتری مشغول به فعالیتم، بیشتر از قبل حس می‌کنم، چرا که بنا به شرایط موجود با آدم‌های بیشتری که برای پروژه انتساب شده‌اند، همکاری دارم. یکی از جدیدترین تجربه‌هام مربوط به زمانیه که برای پروژه‌ی ساخت سخت‌افزار و نرم‌افزار یک ربات ۹ ماه با تیمم در دانشگاه UCSD کار کرده بودم و پروژه از سمت ما آماده‌ی تست بود، اما تیم با یه دانشجو از گروه هنرهای تجسمی هم همکاری می‌کرد که کار تولید پوست برای صورت ربات رو به عهده داشت. این شخص علی‌رغم تعهداتش نتونست کار رو طی زمان مشخص به ما تحویل بده، در نتیجه ما قادر نبودیم تست‌هایی رو که برای سنجش ربات در تعامل با انسان بود به موقع اجرا کنیم و نتایج رو برای چاپ در کنفرانس مورد نظرمون بفرستیم.

گاهی ممکنه شما اشتباهی کنین که در کارتون تاخیر و یا اختلال ایجاد کنه. به طور مشابه، افراد دیگه هم (به احتمال زیاد سهوی) ممکنه اشتباهی بکنن که باعث ایجاد مشکل در کار شما بشه. این امریه که می‌تونه به قدری زیاد روی شما تاثیر بذاره که تمرکز روی کارتون رو از دست بدید و باعث بشه که به صورت احساسی برخورد کنید؛ اما شما باید یاد بگیرید که اشتباهات رو ببخشید، بپذیرید که اتفاق افتادن اشتباه امری محتمل بوده و تمرکز کنید روی روشی برای درست کردن اشتباه اتفاق افتاده. این باعث می‌شه روابط شما با اطرافیان‌تون بهتر بشه و مقدار زمانی که ممکن بود صرف دنبال کردن اشتباه بشه رو به زمان مفید تبدیل کنید و ازش استفاده کنین.

برای پروژه‌ای که براتون مثال زدم، خودم به شخصه وارد جزییات کار هنری پروژه شدم و بعد از چندین جلسه با دانشجوی هنرهای بصری، محدودیت‌های کار رو پیدا کردم و با کمکش گزارشی دقیق از پیشرفت‌ها و اشتباهاتی که طی تولید پوست ایجاد شده بود تهیه کردیم. متوجه شدم نیاز به علوم زیست‌شناسی داریم پس لینک جدیدی با این دپارتمان زدم، دانشجوی جدیدی رو با قابلیت‌های مورد نیاز به تیم اضافه کردم تا توی تولید پوست برای ربات کمک کنه و این بخش کار سرعت بگیره. در آخر به موازات این فرایند شروع به تست سیستم و الگوریتم نرم‌افزاری ربات ساخته شده کردم و الان نتایج تست نرم‌افزار رو برای همون کنفرانس و برای ددلاین تعیین شده آماده کردم. به علاوه برای تیم هنری زمان بیشتری خریدم تا رباتمون رو برای به دست آوردن نتایج تحقیق تعامل ربات با انسان تکمیل کنن. اگر از اون اشتباه به موقع نمی‌گذشتم و به موقع به فکر راه حل نمی‌افتادم، به احتمال قوی نه کار من و نه کار تیم پیشرفت نمی‌کرد.

▪ چی شد به فکر کار در انجمن و شروع نشریه توی دانشکده افتادید؟ اون موقع ترم چند بودید؟

متاسفانه از سال اول احساس کمبود تعاملات علمی خارج از کلاس، فعالیت‌های فرهنگی، آموزشی و گردهمایی‌های علمی پژوهشی رو توی گروهمون حس کردم. اون زمان ۳ نفر عضو انجمن علمی بودن که پایان اون سال تحصیلی هر سه فارغ‌التحصیل شدن و من و دو دوست همیشه همراهم مهندس مهسا شیخ‌حسنی و مهندس مریم مسعودی مسئولیت کارهای انجمن رو به عهده گرفتیم. در ابتدا مسئولیت انجام تمام امور علمی و اداری و فرهنگی و مالی به عهده‌ی ما بود، اما بعد از چندین مرحله تبلیغات در خصوص برنامه‌های انجمن و اهدافی که براش تعیین کرده بودیم، با سازمان‌دهی زیرگروه‌ها و عضوگیری، اعضای انجمن به حدود ۱۵ نفر رسید که هر کدوم بسته به علاقشون در زیرشاخه‌ها شروع به فعالیت کردند.

▪ کی فارغ‌التحصیل شدید؟

ترم هشتم، بهار ۱۳۹۳. ۴ سال درس مهندسی خوندن و فارغ‌التحصیل شدن به اندازه‌ی گرفتن نسخه‌ی فیزیکی مدرک فارغ‌التحصیلی سخت نبود!:))

▪ خاطره‌ای دارید از دوران تحصیل‌تون که بخواین برامون بگید.

توی یک ظهر گرم تابستون ۱۳۹۳ که برای گرفتن مدرک فارغ‌التحصیلی‌ام به دانشگاه رفته بودم، یکی از دانشجوهای سال پایینی رو که اتفاقاً عضو انجمن علمی بود، دیدم. ازم پرسید حالا که فارغ‌التحصیل شدی و داری از این دانشگاه می‌ری چه حسی داری؟ جواب دادم حس نگرانی! نگرانم که نتیجه‌ی کارهایی که توی این سال‌ها انجام دادم، موندگار نباشه و بعد از رفتنم از بین بره. جواب داد «تو انجمنی رو سرپا کردی و نشریه‌ای رو راه انداختی که ما دانشجوهای کامپیوتر بتونیم در اون حرف‌هامون رو بزنیم، تعلیم ببینیم و تعامل بیشتری با جامعه‌ی علمی و فرهنگی داشته باشیم. زحمت‌هایی که تو کشیدی امکاناتی رو برای دانشجوهای این دپارتمان فراهم کرده که قبلاً نداشتند. ما هیچ‌وقت تو و زحمت‌هات رو فراموش نمی‌کنیم.» جمله‌های صمیمانه‌اش جوری به قلبم نفوذ کرد که تا به امروز فراموش نکردم.

اون روز، آخرین روزی بود که به عنوان دانشجوی مهندسی دانشگاه الزهرا (س) به این دانشگاه رفتم و همین‌طور اولین روزی که تاثیر حقیقی سخت‌کوشی‌های چهار سال فعالیت‌های علمی و فرهنگیم در این دانشگاه رو دیدم.

طی سالیان بعد بارها خبرهای خوشی از دوستام و دانشجوهای سال پایینی درباره‌ی پیشرفتی که این گروه‌ها داشتن، بهم رسیده. آخرین خبر که بسیار خوشحالم کرد این بود که نشریه‌ی دانشجویی پردازش توی لیست پنج نشریه‌ی برتر کشوری قرار گرفته. خبرش برای یک هفته انرژیم رو مضاعف کرد!:-)

▪ از اونجایی که هم در ایران و هم در خارج از کشور تحصیل کردید، دیدتون نسبت به دانشگاه‌های ایران چیه؟ این‌که در چه جایگاهی هستن؟ نقاط قوت و ضعف شون چیه؟

به نظرم اصلی‌ترین نقطه قوت دانشگاه‌های ایران، برخورداری از قابلیت و توانمندی و استعداد سرشار نسل جوانه که می‌تونه به مدد همت و تدبیر مسئولین

امر، فضای رشد و شکوفایی پیدا کنه. یکی از مواردی که دانشگاه‌های سطح کشور باید اهتمام بیشتری نسبت به اون داشته باشن، برنامه‌ریزی جهت کشف، شناسایی و پرورش استعدادهاست. استفاده از پتانسیل نیروی جوان در سطح دانشگاه هم می‌تونه برای دانشگاه، ارزش‌افزایی بکنه و برای دانشجوها هم موجب تجربه‌اندوزی و مهارت‌افزایی بشه.

مسئله بعدی تقویت ارتباط صنعت و دانشگاهه. دانشگاه و صنعت می‌تونن پروژه‌های مشترکی رو اجرا کنند. صنعت پتانسیل این رو داره که نیروهای مورد نیازش رو از بین دانشجویان مستعد، جستجو و استخدام کنه. پس چه بهتر که دانشگاه شرایط رو برای تربیت افرادی که مورد نیاز صنعت هستن، فراهم کنه و این امکان رو در اختیار دانشجویان و صنعت‌گران بذاره که با هم ملاقات و نشست‌های حضوری داشته باشن تا شناخت بهتری نسبت به نیازهای طرفین بوجود بیاد.

- **شاغل هستید؟ حوزه فعالیت‌تون رو برامون می‌گین؟**

من در ایالت کالیفرنیا، شهر سن-دیگو زندگی می‌کنم. با شروع دوره‌ی دکتری، کار و منبع درآمدم تحقیق و انجام پروژه‌های علمی پژوهشی، حساب می‌شه. در حال حاضر در گروه contextual robotics institute در دانشگاه UCSD همکاری می‌کنم و ربات‌های انسان‌نما برای کاربردهایی در بهداشت و درمان می‌سازیم.

- **توی ایران هم کار کردید؟ محیط‌های کاری ایران و خارج از کشور چه تفاوت‌هایی دارند؟**

من بلافاصله بعد از فارغ‌التحصیلی از مقطع کارشناسی‌ام اقدامات لازم مثل امتحان تافل و جی ار ای رو انجام دادم و به فاصله‌ی کمی بعد از اون پذیرش گرفتم و برای شروع تحصیلات ارشدم مهاجرت کردم. بعد هم تا به امروز در محیط دانشگاهی مشغول تحصیل و تحقیق بودم، به همین خاطر تجربه کاری قابل ذکری ندارم.

- **ایران در حوزه‌ی کاری ما در جهان چه جایگاهی داره؟ چه پتانسیل‌هایی داره؟ نقاط قوت و ضعفاش چیه؟**

علم کامپیوتر از سرعت پیشرفت بسیار بالایی برخورداره و هر لحظه در حال تغییره. هر روز خبری از یه گوشه از دنیا می‌رسه از نتیجه‌ی تحقیقات فرد یا گروهی که تحولی اساسی در علم روز ایجاد

کرده‌اند. من باور دارم که مهم‌ترین سرمایه‌های یک کشور، سرمایه انسانیه و این سرمایه است که موجب کسب مزیت رقابتی می‌شه. کشور ما سرشار از سرمایه انسانیه. کافیه این سرمایه‌ها را بیشتر دریابیم و فضای رشد و پرورش اون‌ها رو در داخل کشور فراهم کنیم تا بیش از پیش شاهد بالندگی نخبگان کشورمون در جای‌جای جهان باشیم. یکی از مناسب‌ترین روش‌ها برای این منظور می‌تونه افزایش دسترسی دانشجوها به منابع علمی و مقالات به روز دنیا باشه که شرایط رو برای آگاهی از علم روز و رقابت آگاهانه برای پیشرفت‌های علمی فراهم می‌کنه.

- **در آمریکا چجوریه؟ بانوان در مهندسی کامپیوتر چقدر فعال و تاثیرگذارهستند؟ آیا اونجا هم بین بانوان و آقایون در این حوزه تفاوتی هست؟**

به نظر می‌رسه تبعیض جنسیتی، به‌خصوص در فضاهای کاری مرتبط با علوم کامپیوتر امری شناخته شده در بیشتر کشورهای دنیاست که بسیاری از متفکرین (اعم از زن و مرد) در حال تلاش برای اصلاحش هستن. برای مثال ماه آینده توی تگزاس آمریکا بیش از ۲۰،۰۰۰ نفر در گردهمایی سالانه‌ای با محوریت زنان در علم کامپیوتر و تکنولوژی جمع می‌شن تا تعلیم‌های لازم برای برابری جنسیتی و نحوه‌ی افزایش موقعیت‌های علمی و شغلی برای بانوان رو آموزش و گسترش بدن.

با این اوصاف جهان داره به سمت و سویی می‌ره که توانمندی و مهارت فرد، تعیین کننده‌تر و پررنگ‌تر از مسائل جنسیتی می‌شه. این مسئله خاص کشوری هم نیست. افراد می‌تونن در هر حوزه‌ای حرفی برای گفتن داشته باشن به شرط این‌که به دانش روز اون حوزه اشراف داشته باشن. به همین خاطر یه راه معقول اینه که خودمون رو باور داشته باشیم و بدونیم هرکس که توانمند و در عین حال خلاق باشه با تلاش می‌تونه به موفقیت برسه. خانم دکتر نوشین ریاحی که در زمان دانشجویی من مدیر گروه کامپیوتر دانشگاه بودن، نمونه‌ی خوبی از بانوان کشور در مهندسی‌کامپیوتر هستن که سازماندهی و نظم‌شون تاثیر بسیار مثبتی در روند تحصیلی دانشجوها گذاشته.

- **اگر برگردید به عقب همین مسیر رو انتخاب می‌کنید؟ (چرا؟)**

همیشه به خودم می‌گم هدف خوب داشتن مهمه و باید حواسمون باشه انرژی خودمون رو صرف کارهای مفید کنیم. فکر می‌کنم تا به این لحظه انرژی و استعدادهام رو صرف راه و هدف درستی کردم، برای همین قطعاً اگر به عقب برگردم بازم همین رشته رو انتخاب می‌کنم.

- **نصیحت و پیشنهادی به بچه‌ها**

به تمام دوستانی که قدم توی مسیر علم‌آموزی و یا علم‌پروری گذاشتن، پیشنهاد می‌کنم بیشتر به ندای درونی‌تون گوش کنید؛ استعدادهاتون رو شناسایی و تقویت کنید. از جمله مهارت‌های مهم در دنیای امروز، خلاقیت و قدرت حل مسئله هست، مهارت‌هایتان را افزایش دهید. برای خودتون هدف‌های کوتاه‌مدت و بلندمدت تعیین کنید و برای رسیدن بهشون برنامه‌ی مرحله به مرحله بریزید.

پدر، مادر و یا هرکسی که شما رو بزرگ کرده بیشترین شناخت رو نسبت به شما داره، ضعف‌ها و قوت‌هاتون رو می‌دونه. حرف‌هاشون رو بشنوین و اگر بهتون توصیه‌ای کردن بهش فکر کنین و اگر تونستین بهش عمل کنین.

- **سوالی بود که دوست داشتید بپرسیم و ما نپرسیدیم؟**

فکر کنم این طولانی‌ترین مصاحبه چند سال اخیرتون باشه، امیدوارم نکاتی که تا اینجا گفتم برای خواننده‌های مجله جالب و مفید بوده باشه.:)

- **در نهایت صحبت دیگه‌ای دارید برامون؟**

در آخر این‌که:

بهترین‌ها رو برای جوانان سرزمینم آرزومندم.

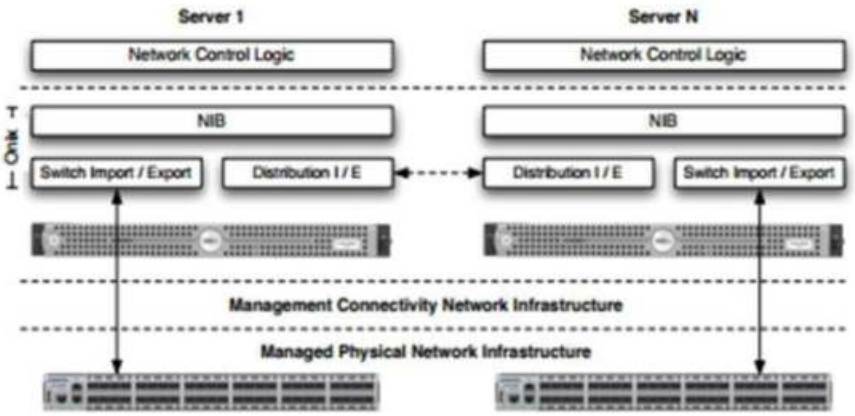
آشنایی با SDN (قسمت دوم)

۷-کنترل‌کننده Onix

بااینکه تاکنون سیستم‌هایی متعددی ارائه شده‌اند که از الگوی اصلی SDN پیروی می‌کنند، اما هنوز مطالب کمی منتشر شده که نشان دهد چگونه پایگاه کنترلی بسازیم که به همه الزامات و نیازها پاسخگو باشد. برای پر کردن این شکاف، سیستم‌عامل Onix معرفی شد.

همان‌طور که در شکل۶ می‌بینید، چهار جز در یک شبکه کنترل Onix وجود دارد: زیربنای فیزیکی مدیریت شده، زیربنای اتصال، Onixو منطق کنترل که توسط اپلیکیشن‌های مدیریتی به کار گرفته شده است. این شکل دو نمونه‌ی هماهنگ Onix را نشان می‌دهد که دیدشان را از وضعیت شبکه به اشتراک می‌گذارند و به منطق کنترل یک واسط خواندن/نوشتن به وضعیت پیشنهاد می‌کنند (با فلش تیره نشان داده شده است) .

مهم‌ترین فایده Onix برای کاربردهای فعلی به دو بخش تقسیم می‌شود: اول، Onixنسبت به سیستم‌های قبلی API عمومی‌تری را در اختیار قرار می‌دهد؛پروژه‌هایی که بر روی Onix ساخته می‌شوند محیط‌هایی به گوناگونی WAN را هدف قرار می‌دهند. ابر عمومی (public cloud) و مرکز داده شرکت هم از محیط‌های هدف آن است. دوم، Onix اصول ابتدایی توزیع منطقی را فراهم می‌کند(مثل حافظه DHT و عضویت گروه)که طراحان اپلیکیشن را قادر می‌سازد بدون اینکه مکانیزم‌های توزیع را بازسازی کنند، اپلیکیشن های کنترل را به کار بگیرند و انعطاف‌پذیری را طوری نگه دارند تا تبادلات عملکرد/ مقیاس پذیری همانند آنچه الزامات اپلیکیشن دیکته می‌کند انجام شوند.



شکل ۶: اجزاOnix

برای درک اینکه چگونه Onix به یک پایگاه کنترل تولید- کیفیت تحقق می‌بخشد لازم است درباره دو جنبه طراحی آن بحث کنیم: بستری که Onixدر قالب آن در شبکه قرار می‌گیرد و API ای که Onix در دسترس طراحان اپلیکیشن قرار می‌دهد.

۱-۷-اجزا

طبق شکل ۶ شبکه‌ای که با Onix کنترل شود از چهار جز تشکیل می‌شود که هرکدام نقش متمایزی دارند.

✓ زیربنای فیزیکی: شامل سوئیچ‌ها و مسیریاب‌های شبکه است که واسطی را پشتیبانی می‌کنند که آن واسط Onix را قادر می‌سازد وضعیتی را بخواند و بنویسد که رفتار عناصر (مثل درایه‌های موجود در جدول انتقال) را کنترل می‌کند. این عناصر نیازی به اجرای هیچ نرم‌افزار دیگری ندارند .

✓ زیربنای اتصال: ارتباط بین شبکه فیزیکی و Onix (ترافیک کنترل) انتقال را در اتصالات زیرساخت کنترل می‌کند. این کانال کنترلی هم می‌تواند به‌صورت In-band باشد (که در آن ترافیک کنترل همان عناصر ارسالی را به اشتراک می‌گذارد که ترافیک داده در شبکه به اشتراک می‌گذارد) و هم می‌تواند به‌صورت Out-bandباشد (که در آن‌یک شبکه فیزیکی مجزا برای اداره ترافیک کنترل استفاده می‌شود). زیربنای اتصال باید ارتباط چند وجهی بین نمونه‌های Onix و سوئیچ‌ها برقرار کند و در صورت لزوم از همگرایی در خطای Link پشتیبانی کند. پروتکل‌های مسیریابی استاندارد) مثل IS-IS یا (OSPF برای ساخت و نگه داری وضعیت ارسال در زیربنای اتصال مناسب هستند .

✓ Onix: یک سیستم توزیع شده است که در خوشه‌ای از یک یا چند سرور فیزیکی اجرا می شود. هر سرور هم می‌تواند شامل چند نمونه از سیستم‌های Onix باشد .Onix به عنوان پایگاه کنترل موظف است به منطق کنترل توانایی برنامه ریزی شبکه را بدهد (هم نوشتن و هم خواندن وضعیت شبکه). یک نمونه Onix همین‌طور برای منطبق شدن با مقیاس شبکه‌های بزرگ (با میلیون ها پورت) و برای فراهم کردن قابلیت برگشت پذیری ضروری برای توسعه تولید، موظف است

وضعیت شبکه را به نمونه‌های دیگر درون خوشه ارسال کند.

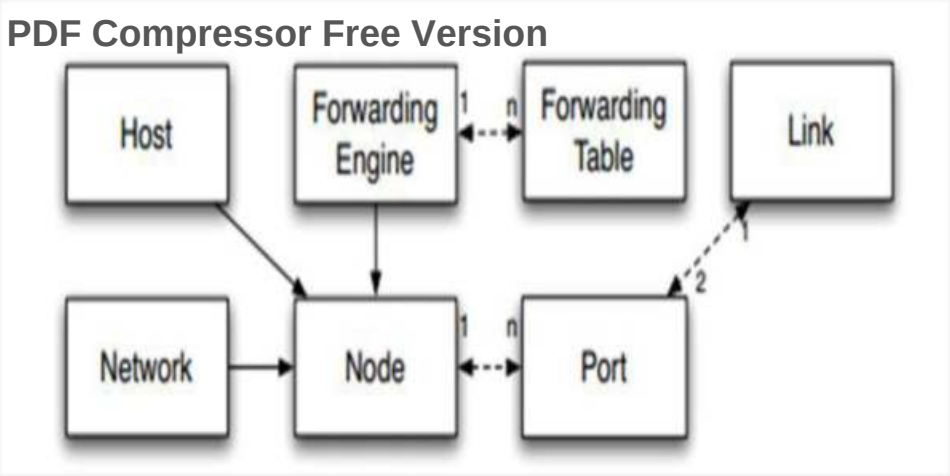
✓ منطق کنترل: منطق کنترل شبکه در رأس API اونیکس قرار دارد. این منطق کنترل رفتار مطلوب و مورد انتظار شبکه را تعیین می‌کند؛ Onix صرفاً اصول ابتدایی لازم برای دستیابی به وضعیت مناسب شبکه را فراهم می‌کند.

این‌ها چهار جز اصلی شبکه مبتنی بر SDN هستند. فرض می‌کنیم که زیربنای فیزیکی می‌تواند بسته‌ها را بسیار سریع (معمولاً دو یا چند برابر سریع‌تر) انتقال دهد. سریع‌تر از آن‌که Onix (یا هر پایگاه عمومی کنترل دیگری) بتواند آن را پردازش کند. از این نظر استفاده از Onix برای اجرای عملکردهای مدیریتی مناسب نیست. عملکردهایی که برای شناخت تغییرات در وضعیت شبکه به ازای هر بسته (یا هر تغییر سریع دیگری) نیاز به منطق کنترل دارند.

API-7-2

کمک اصلی Onix تعریف یک API مفید و عمومی برای کنترل شبکه است که توسعه اپلیکیشن‌های مقیاس پذیر را امکان‌پذیر سازد. در ادامه‌ی کارهای قبلی، API و انیکس را حول یک شبکه فیزیکی طراحی شده است تا اپلیکیشن‌های کنترل را قادر باشند وضعیت را برای هر عنصر در شبکه بخوانند یا بنویسند. از این رو این API با محوریت داده، روش‌هایی را برای پایدار نگه داشتن شبکه بین عناصر درون شبکه‌ای و اپلیکیشن کنترلی) که بر روی نمونه‌های متعددی از Onix قابل اجرا هستند (ارائه می‌کند. به طور خاص، API اونیکس شامل یک مدل داده می‌باشد که زیربنای شبکه را با هر عنصر شبکه که به یک یا چند حامل داده مربوط می‌شود نشان می‌دهد. منطق کنترل می‌تواند: وضعیت فعلی مربوط به آن حامل داده را بخواند، با کار بر روی این حاملان وضعیت شبکه را تغییر دهد و گزارش‌های تغییر وضعیت این حاملان را ثبت کند. علاوه بر این، از آن جا که Onix باید طرح‌های کنترلی بسیاری را پشتیبانی کند، پایگاه به منطق کنترل اجازه می‌دهد مدل داده را تغییر دهد و بر استقرار و پایداری هر جز شبکه کنترل داشته باشد.

یک کپی از وضعیت شبکه که Onix مسیریابی می‌کند در یک ساختمان داده ذخیره می‌شود. این سازه داده‌ای مبنای اطلاعات شبکه



شکل ۷: کلاس‌های موجودیت‌های پیش فرض شبکه ارائه شده توسط API اونیکس

(NIB)نام دارد که بسیار شبیه به مبنای اطلاعاتی مسیریابی (RIB) است که مسیریاب‌های IP از آن استفاده می‌کنند. NIB هم پیشوندها را در مقاصد ذخیره می‌کند و هم گرافی از همه عناصر موجود در توپولوژی شبکه دارد. NIB قلب مدل کنترلی Onixاست و مبنایی است برای مدل توزیع شده .Onixاپلیکیشن‌های کنترل شبکه می‌توانند در NIBبنویسند و از آن بخوانند (برای مثال حالت ارسال را اصلاح می‌کنند یا به شمارنده‌های پورت دستیابی پیدا می‌کنند) و Onix با تکرار و توزیع NIBبین موارد مختلف در حال اجرا قابلیت مقیاس پذیری و برگشت پذیری را (در چارچوبی که اپلیکیشن معین کرده) فراهم می‌کند.

Onix در زمانی که تکرارها و توزیع داده‌های NIB را اداره می‌کند، برای یافتن و فراهم کردن راه حل برای تضادها در وضعیت شبکه به منطق مخصوص اپلیکیشن متکی است. تضاد در وضعیت شبکه حین تبادل داده بین نمونه‌های Onixو همین طور بین یک نمونه Onix و یک عنصر شبکه روی می‌دهد. منطق کنترل می‌تواند ضمانت‌های سازگاری را برای وضعیتی که بین نمونه‌های Onix با استفاده از کدگذاری توزیع شده و الگوریتم‌های همه پذیر پخش شده است دیکته کند.

برای ساده کردن موضوع، ما فرض می‌کنیم NIB تنها موارد فیزیکی موجود در شبکه را شامل می‌شود. اگرچه از نظر عملی به سادگی می‌توان آن را برای پشتیبانی عناصر منطقی (مثل کانال‌ها) توسعه داد.

7-3-جزئیات مبنای اطلاعاتی شبکه

به طور کلی، NIB مجموعه‌ای از موارد موجود در شبکه را در برمی‌گیرد که هر کدام دارای مجموعه‌ای از زوج‌های key-value است و با یک شناساگر جهانی مسطح ۱۲۸ بیتی شناسایی می‌شود. این عناصر موجود در شبکه ساختار پایه را تشکیل می‌دهند و همه انواع دیگر موجود در شبکه از آن ناشی می‌شوند Onix.دسته بندی قوی‌تری را برای عناصر دسته بندی شده پشتیبانی می‌کند. سپس انواع دسته‌بندی شده مجموعه‌ای از پیش تعیین شده از ویژگی‌ها و روش‌ها را شامل می‌شوند)با استفاده

Category	Purpose
Query	Find entities.
Create, destroy	Create and remove entities.
Access attributes	Inspect and modify entities.
Notifications	Receive updates about changes.
Synchronize	Wait for updates being exported to network elements and controllers.
Configuration	Configure how state is imported to and exported from the NIB.
Pull	Ask for entities to be imported on-demand.

7-3-جزئیات مبنای اطلاعاتی شبکه

از زوج (key-value و بر اساس این ویژگی‌ها عمل می‌کنند.

برای مثال، یک کلاس موجودیت Port وجود دارد که می‌تواند به لیستی از پورت‌ها در موجودیت Node متعلق باشد. شکل ۷ مجموعه اصلی موجودیت‌های دسته بندی شده را نشان می‌دهد که Onix ایجاد می‌کند –موارد موجود در هر دسته در یک کلاس پایه‌ای مشترک محدود به دسترسی عمومی زوجkey-value قرار می‌گیرند. چیدمان دسته‌ها در Onix ثابت نیست و اپلیکیشن‌ها می‌توانند این کلاس‌های اصلی را به کلاس‌های ریزتری دسته بندی کنند تا در صورت لزوم مدل داده Onix را توسعه دهند.

در شکل۷، خطوط توپر نشان دهنده‌ی ارث بری و خط چین‌ها مربوط به ارتباط ارجاعی بین نمونه‌های موجودیت‌ها می‌باشند. اعداد روی خط‌چین‌ها نمایانگر رابطه‌های نگاشت کمی هستند (برای مثال یک لینک به دو پورت نگاشت می‌شود و دو پورت می‌توانند به همان لینک نگاشت شوند). گره‌ها، پورت‌ها و لینک‌ها توپولوژی (وضعیت مکانی) شبکه را شکل می‌دهند. تمامی کلاس‌های موجودیت از همان کلاس پایه‌ای ارث می‌برند که دسترسی زوج key-value را فراهم می‌کند.

NIB برای اینکه منطق کنترل بتواند به موجودیت‌های شبکه دسترسی یابد، راه‌های مختلفی را در اختیار می‌گذارد. همچنین فهرستی از تمام موجودیت‌ها بر اساس شناساگر آن‌ها تهیه و نگه‌داری می‌کند تا جستجوی مستقیم برای یک موجودیت خاص را امکان پذیر سازد. علاوه بر این از ثبت گزارش‌های تغییر وضعیت یا اضافه/حذف شدن موجودیت‌ها پشتیبانی می‌کند. اپلیکیشن‌ها بعداً می‌توانند با توجه به گزارش‌های ورودی یک

موجودیت در شبکه و نگه‌داری فهرست خودشان، قابلیت جستجو را توسعه دهند.

همه عملکردهای NIB ناهماهنگ (اسنکرون) هستند. به این معنی که به روز کردن یک موجودیت در شبکه فقط این را تضمین می‌کند که پیام به روز رسانی بالاخره به عنصر مربوطه در شبکه و/یا دیگر نمونه‌های Onix ارسال خواهد شد، اما تأخیر تضمین نمی‌شود.

۸-مقیاس پذیری و قابلیت اطمینان

Onix برای اینکه بتواند جایگزینی برای ساختار سنتی شبکه باشد، باید به الزامات مقیاس پذیری و قابلیت اطمینان شبکه‌های امروز و آینده پاسخگو باشد. از آن جا که NIB نقطه مرکزی وضعیت سیستم و رویدادها محسوب می‌شود، استفاده از آن در سطح بزرگ مقیاس پذیری و قابلیت اطمینان سیستم را مشخص می‌کند. برای مثال، همزمان با کاهش تعداد عناصر شبکه، NIB ای که توزیع نشده باشد باعث از بین رفتن حافظه سیستم می‌شود. یا تعداد رویدادهای شبکه)رویداهایی که NIB ایجاد کرده (یا میزان کاری که برای مدیریت آن‌ها باید انجام شود زیاد می‌شود و CPUیک نمونه واحد Onix را اشباع می‌کند.

۸.۱-مقیاس پذیری

Onix از سه راهبرد برای بهبود مقیاس پذیری استفاده می‌کند: اول، اپلیکیشن‌های کنترلی را قادر می‌سازد بارکاری را تفکیک کنند. از این رو اضافه کردن نمونه‌ها بار کاری را کم می‌کند بدون اینکه صرفا آن را تکرار کند. دوم، Onix تراکم را امکان پذیر می‌سازد. در تراکم، شبکه‌ای که با خوشه‌ای از گره‌های Onix مدیریت می‌شود به عنوان یک گره واحد در NIB برای یک خوشه مجزا دیده می‌شود. این امر دسته‌های موجود در Onix را متحدالشکل و ساختار آن را سلسله مراتبی می‌کند. از این رو باعث می‌شود میزان اطلاعات مورد نیاز درون یک خوشه واحد Onix کاهش یابد. در نهایت، Onix اپلیکیشن‌هایی را در دسترس قرار می‌دهد که بر روی پایداری و دوام وضعیت شبکه کنترل دارند. به طور کلی:

تفکیک:(Partitioning) منطق کنترل شبکه می‌تواند Onix را طوری پیکربندی کند که یک نمونه کنترلر به خصوص تنها یک زیرمجموعه از NIB را در حافظه نگه داری کند و آن را به روز نگه دارد. علاوه بر این، یک نمونه Onix می‌تواند اتصالاتی به یک زیرمجموعه از عناصر شبکه داشته باشد تا تعداد رویدادهایی که عناصر ایجاد می‌کنند و این نمونه باید پردازششان کند کمتر شود.

متراکم سازی:(Aggregation) در تنظیمات متشکل از چندOnix ، یک نمونه از Onixمی‌تواند گروهی از عناصر درون NIB خود را به عنوان یک عنصر متراکم به یک نمونه Onix دیگر نشان دهد. این امر معمولاً برای نشان دادن پیچیدگی کمتر به لایه‌های بالایی در سلسله‌ای از کنترلرهای Onix انجام می‌شود. برای مثال، در یک محوطه بزرگ شبکه، هر ساختمان با یک کنترلر) Onixیا خوشه‌ای از کنترلرها) مدیریت می‌شود. این کنترلر تمام عناصر موجود در آن ساختمان را به عنوان یک گره متراکم به یک نمونه جهانی Onixکه وظیفه مهندسی ترافیک بین محوطه‌ها را به عهده دارد نشان می‌دهد. این موضوع مشابه مفهوم الگوهای مدیریت جهانی کنترل در شبکه‌های ATM است.

سازگاری و دوام (Consistency and durability): منطق کنترل الزامات پایداری را به وضعیت شبکه‌ای که آن را مدیریت می‌کند دیکته می‌کند. این کار با پیاده سازی هر کدام از الگوریتم‌های سازگاری مورد نیاز برای رسیدن به وضعیت سازگار انجام می‌پذیرد. برای وضعیت‌هایی که در آن استفاده از این الگوریتم‌ها، سازگاری را تضمین نمی‌کند، مسیریابی و تفکیک ناسازگاری به ایجاد پایداری کمک می‌کند. به طور پیش فرض، Onixدو نوع پایگاه ذخیره داده ارائه می‌کند که اپلیکیشن می‌تواند از آن‌ها برای حالاتی استفاده کند که ترجیحات متفاوتی برای سازگاری و دوام دارند. برای اپلیکیشن‌هایی که به دنبال دوام و سازگاری قوی‌تری هستند، Onix یک پایگاه داده تراکنشی تکرار شونده پیشنهاد می‌کند و برای وضعیت‌های فرار که تحملشان در برابر عدم پایداری بیشتر است یکDHT تک هاپ مبتنی بر حافظه ارائه می‌کند.

مکانیزم‌های مقیاس پذیری فوق می‌توانند برای مدیریت شبکه‌هایی استفاده شوند که بسیار بزرگ‌تر از آن هستند که تنها با یک نمونه‌ی Onix کنترل شوند. برای نشان دادن این موضوع، از یک مثال کاربردی استفاده می‌کنیم؛ اپلیکیشنی را در نظر بگیرید که می‌تواند بین سوئیچ‌ها در توپولوژی مدیریت شده مسیر ایجاد کند، با این هدف که مسیرهای کاملی درون شبکه به وجود آورد.

تفکیک: شبکه‌ای را فرض می‌کنیم که تعدادی متعادل از سوئیچ‌ها دارد و می‌تواند به سادگی توسط یک نمونه Onix اداره شود. درنتیجه، تعداد و سایز همه ورودی‌های ارسالی وضعیت بر روی شبکه، منابع حافظه یک سرور فیزیکی واحد را اشغال می‌کنند.

برای رفع چنین مشکلی، منطق کنترل می‌تواند وضعیت تمام سوئیچ‌ها را تکرار کند، اما باید وضعیت ارسال را تفکیک کند و هر بخش را به یک نمونه منحصر به فرد Onix اختصاص دهد تا آن نمونه وظیفه مدیریت آن بخش را به عهده بگیرد. تا زمانی که تفکیک بتواند قطعه‌های نسبتاً پایدار ایجاد کند.

منطق کنترل می‌تواند اطلاعات سوئیچ و لینک را در یک وضعیت کاملاً سازگار و با دوام و مشترک بین تمام نمونه‌های Onix ضبط کند. همچنین می‌تواند با استفاده از مکانیزم‌هایی که چارچوب در دسترس قرار داده به روز رسانی‌ها را هماهنگ کند. هرچند، اطلاعاتی که فراترند، مثل سطوح کاربری لینک‌ها، می‌توانند در DHT ذخیره شوند. هر کنترلر می‌تواند از ارائه توپولوژی فیزیکی کامل) NIB از پایگاه داده سازگار) که با داده‌های کاربری لینک) از (DHT همراه است استفاده کند. به این صورت کنترلر وضعیت ارسالی را شکل می‌دهد که برای تحقق بخشیدن به الزامات توسعه در سرتاسر شبکه، تضمین راه‌ها در آن ضروری است.

متراکم سازی: زمانی که شبکه مورد نظر رشد زیادی کند، تفکیک در مدیریت مسیر دیگر کافی نیست. فرض می‌کنیم هنوز نمونه‌های Onix قادر به دربرداشتن NIB کامل هستند، اما منطق کنترل با وجود تعداد بالای رویدادهای شبکه نمی‌تواند به کار خود ادامه دهد و از این رو CPU را اشباع می‌کند.CPU عامل محدود کننده معمول برای اپلیکیشن‌های کنترلی است.

برای محافظت نمونه‌های راه دور از نرخ‌های بالای به روز رسانی، این اپلیکیشن می‌تواند یک توپولوژی را متراکم سازد و به شکل یک گره واحد دربیاورد و از آن به عنوان واحد انتقال رویداد بین نمونه‌ها استفاده کند. برای مثال، توپولوژی می‌تواند به نواحی منطقی تقسیم شود که هر کدام توسط یک نمونه مجزای Onix مدیریت شود. نمونه‌های Onixخارج از یک ناحیه، توپولوژی فیزیکی دقیق درون ناحیه را می‌شناسند اما فقط اطلاعات کاربری لینکی را که از نظر وضعیت جغرافیایی متراکم شده است از) DHT که در اصل از نمونه‌های درون همان ناحیه ایجاد شده‌اند) بازیابی خواهند کرد.

تفکیک بیشتر: در برخی موارد، تعداد عناصر درون یک قلمرو کنترل، ظرفیت یک نمونه واحد Onix را اشغال می‌کند. اگرچه به علت نرخ‌های تغییر نسبتاً آهسته در شبکه فیزیکی، همچنان بدست آوردن یک دید توزیعی بر گراف شبکه (NIB) امکان پذیر است.

اپلیکیشن ها می‌توانند همچنان بر متراکم سازی اطلاعات لینک‌های کاربری متکی باشند، ولی در یک برنامه NIB تفکیک شده، از مکانیزم های توزیع وضعیت میان Onix ی برای ردوبدل درخواست‌ها به سوئیچ‌ها در ناحیه‌های راه دور استفاده کرد. اگر از ویژگی‌های NIB به عنوان کانال ارتباط راه دور استفاده کنیم، این کار انجام شدنی است. "درخواست" و "پاسخ" با استفاده از DHT بین ناحیه‌ها مخابره می‌شوند. از آن‌جا که این نقل و انتقال می‌تواند توسط یک نمونه سوم Onix انجام شود، هر اپلیکیشنی که "پاسخ"ها را سریع‌تر بخواهد باید حدود کلیدها در DHT برای نواحی را بشناسد و از کلیدهای DHT طوری استفاده کند که برای ورودی معین، ویژگی‌های آن در ناحیه درست ذخیره شده باشد.

این رویکرد می‌تواند آرایش یک ناحیه گسترده را امکان پذیر سازد. برای مثال، هر قسمتی که تفکیک شده است، می‌تواند نمایانگر یک ناحیه بزرگ شبکه باشد و به خوشه‌ای از نمونه‌های Onix متشکل از گره‌های متراکم که برای مسیریابی‌ها به طور سراسری تصمیم می‌گیرند، هر شبکه به صورت یک گره متراکم نشان داده شود. بنابراین، هر قسمت برای مسیریابی داخلی تصمیم‌گیری می‌کند و خوشه‌ها به تصمیم‌گیری مسیریابی میان این قسمت‌ها می‌پردازند (هر کدام را به عنوان یک گره منطقی مجزا در نظر می‌گیرند.) .

متراکم سازی میان‌domainی: وقتی شبکه کنترل شده دو AS مجزا را پوشش می‌دهد، به اشتراک گذاشتن اطلاعات کامل توپولوژی بین نمونه‌های Onix غیرممکن می‌شود. به خاطر اینکه دلایل امنیتی و طراح منطق کنترل نیاز دارند دوباره طراحی را با نیازهای متغیر سازگاری دهند.

چارچوب، چگونگی نظیر شدنASها را مشخص نمی‌کند ولی در یک سطح بالا برای تحقق باید به دو الزام پاسخگو باشد: ۱- به اشتراک گذاشتن اطلاعات توپولوژی آن‌ها با نظیرشان با میزانی از جزئیات (و در عین حال حفظ امنیت) و ۲- ایجاد مسیر برای یکدیگر به طور فعالانه یا بر حسب لزوم و تقاضا (با توجه به ارتباط با نظیر) و تبادل اطلاعات اجازه ورود .

۸،۲- قابلیت اطمینان

اپلیکیشن‌های کنترلی روی Onix باید چهار نوع خطای شبکه را رفع و رجوع کنند؛ خطای ارسال عناصر، خطای لینک، خطای نمونه Onixو خطا در اتصال بین عناصر شبکه و نمونه (Onixو بین خود نمونه‌های Onix) در این بخش به ترتیب هر کدام را توضیح می‌دهیم.

خطای عناصر شبکه و لینک: صفحه‌های کنترلی مدرن مسئول رفع خطای عناصر شبکه و لینک هستند و منطق کنترل ساخته شده در Onixمی‌تواند از مکانیزم‌های مشابه استفاده کند. اگر در یک عنصر شبکه یا لینک خطا بروز کند، دفعات انتشار خطا درون شبکه همراه با بازشماری جداول ارسالی، حداقل زمان واکنش به خطا را تعیین می‌کنند. وقتی زمان همگرایی را به الزامات دقیقی که وجود دارند اضافه کنیم، می‌توان ترجیح داد که بخشی از همگرایی توسط راه‌های) backup بازیابی) با مکانیزم‌های سریع مواجهه با خطا در عنصر شبکه اداره شود .

خطایOnix : منطق کنترل برای رفع خطایOnix دو گزینه دارد؛ نمونه‌های فعال می‌توانند گره‌ای را که خطا دارد پیدا کنند و سریعاً مسئولیت‌های آن نمونه را به عهده بگیرند یا اینکه بیشتر از یک نمونه می‌توانند به طور همزمان هر عنصر شبکه را مدیریت کنند.

Onix امکانات هماهنگی را فراهم می‌کند تا گره مورد نظر پیدا شود و به خطای نمونه واکنش نشان داده شود. برای مدیریت همزمان یک عنصر شبکه توسط بیشتر از یک نمونه‌ی Onix، منطق کنترل موظف است شرایط گم‌شده نرخ بروزرسانی را همگام با نوشتن در وضعیت شبکه برطرف کند. برای کمک، اونیکس hook هایی فراهم می‌کند که اپلیکیشن‌ها می‌توانند استفاده کنند تا تشخیص دهند آیا تغییرات متناقض که توسط دیگر نمونه‌ها روی عنصر شبکه به وجود آمده است می‌تواند باطل شوند یا نه. با فراهم شدن این امکان، منطق کنترل همان

وضعیت عنصر شبکه را به طور جبری در هر نمونه Onix محاسبه می‌کند. یعنی هر نمونه Onixالگوریتمی یکسان را به کار می‌گیرد. در سطح بالا، این رویکرد مشابه مکانیزم قابلیت اطمینان RCP است که در آن کنترلرهای متمرکز متعدد، به روز رسانی‌ها را از iBGP به روترهای لبه هدایت می‌کنند.

خطای زیرساخت اتصال: مکانیزم‌های توزیع وضعیت Onix خود را از توپولوژی متضمن جدا می‌کنند. از این رو، برای بازیابی و بهبود پس از خطا به اتصال نیاز دارند. اتصال هم بین عناصر شبکه و نمونه‌های Onix و هم بین خود نمونه‌های .Onix برای برقراری این اتصال روش‌هایی وجود دارد. برخی از روش‌های رایج تر را در ادامه شرح می‌دهیم.

برای یک شبکه عملیاتی، داشتن یک شبکه فیزیکی یا VLAN برای مدیریت غیرعادی نیست. برای مثال در ساختار یک پایگاه داده بزرگ. در چنین محیطی Onix می‌تواند از شبکه مدیریتی ترافیک کنترل استفاده کند و آن را از قطع شدن ارتباط صفحه ارسالی مصون بدارد. تحت این مدل استقرار شبکه کنترل از شبکه استاندارد استفاده می‌کند و بنابراین هر گونه قطع در شبکه کنترل توسط پروتکل‌های سنتی رفع می‌شود .(برای مثال OSPF یا درخت پوشا)

حتی اگر محیط، یک شبکه کنترل مجزا در دسترس قرار ندهد، Onix معمولاً بر توپولوژی شبکه فیزیکی اشراف دارد. از این رو، برای منطق کنترل این امکان وجود دارد برای ایجاد اتصال بین Onix و سوئیچ‌ها، عناصر شبکه را با وضعیت ارسالی ایستا مستقر کند. برای تضمین اتصال در مواقع بروز خطا، مسیریابی منبع می‌تواند با ایجاد راه‌های چندگانه ترکیب شود (که در Onix هم بکار گرفته شده است)؛ بسته‌های مسیریابی منبع در راه‌های چندگانه می‌توانند اتصال با قابل اطمینان بالا را به عناصر مدیریت شده شبکه و همین طور بین نمونه‌هایOnix تضمین کنند.

۹-توزیعNIB

در این بخش توضیح می‌دهیم چگونه Onixپایه اطلاعات شبکه خود (NIB) را توزیع می‌کند و معنای سازگاری‌ای که یک اپلیکیشن می‌تواند ازOnix انتظار داشته باشد چیست.

۹-1- کلیات

حمایت Onix از مکانیزم‌های توزیع وضعیت توسط دو ملاحظه بر روی اپلیکیشن‌های مدیریت شبکه هدایت شده است. اول اینکه، اپلیکیشن‌ها از نظر مقیاس پذیری، فراوانی به روز رسانی‌ها در فضای مشترک و پایداری، الزامات متفاوتی دارند. برای مثال سیاست شبکه به کندی تغییر می‌کند و الزامات پایداری جدی و دقیقی دارد. بر عکس، منطقی که از اطلاعات بار لینک استفاده می‌کند بر شبکه‌ای متکی است که وضعیتش با سرعت بالا تغییر کند و طبیعتاً ناپایدارتر است (و بنابراین آن الزامات پایداری را نخواهد داشت).

دوم اینکه، اپلیکیشن‌های متمایز معمولاً برای وضعیت شبکه‌ای که آن را مدیریت می‌کنند الزامات متفاوتی دارند. اطلاعات وضعیت لینک و تنظیمات سیاست شبکه دو نمونه مثال زدنی در این زمینه هستند. اداره کردن پرچم‌های وضعیت موقتی و ناپایدار در لینک‌های مجاور برای اپلیکیشن آسان‌تر از رفع یک ناپایداری در سیاست شبکه است. بنابراین برای این مورد، نیاز است یک متخصص انسانی حضور داشته باشد تا راه‌حل را به طور صحیح پیاده کند.

Onix با ارائه دو مکانیزم مجزا برای توزیع به روزرسانی‌های وضعیت شبکه بین نمونه‌های Onix، از توانایی انتخاب اپلیکیشن بین پایداری و سرعت به روزرسانی‌ها حمایت می‌کند؛ یکی برای نرخ‌های بالای به روزرسانی طراحی شده و در دسترس‌تر است و دیگری با درنظر گرفتن دوام و پایداری طراحی شده است. در ادامه مثال سیستم‌های حافظه توزیع شده که اپلیکیشن‌ها را قادر می‌سازد تبادلات عملکرد/مقیاس پذیری انجام دهند. Onix انتخاب مکانیزم مطلوب برای هر وضعیت داده شده در NIB را به طراحان اپلیکیشن‌ها می‌سپارد و مسئولیت وضوح و صراحت را به آن‌ها واگذار می‌کند. طراحان می‌توانند NIB را به تنهایی به عنوان حافظه برای وضعیت داخلی استفاده کنند. علاوه بر این، اگر اپلیکیشن‌ها ماژول‌های import و exportخود را بنویسند، Onix می‌تواند سیستم‌های حافظه اختیاری را هم پشتیبانی کند. این ماژول‌ها به ترتیب داده را از سیستم حافظه به NIBو از NIB به سیستم حافظه منتقل می‌کنند.

۹.۲- توزیع وضعیت بین نمونه‌هایOnix

Onixاز مکانیزم‌های متفاوتی برای نگهداری ثبات وضعیت بین نمونه‌هایش و بین خود و عوامل ارسال شبکه استفاده می‌کند. دلایل این موضوع دو قسمت دارد: اول، سوئیچ‌ها عموماً CPU های مدیریتی کم قدرت و حافظه‌های RAM محدود دارند. بنابراین پروتکل باید کم وزن و عمدتاً برای ثبات وضعیت ارسال باشد. از طرف دیگر، نمونه‌های Onixمی‌توانند بر روی پایگاه‌های محاسبه عمومی پرقدرت کار کنند و چنین محدودیت‌هایی نداشته باشند. دوم، پیش نیازهای لازم برای مدیریت وضعیت سوئیچ محدودتر است و نسبت به نیازهای دیگر اپلیکیشن‌ها بهتر تعریف شده است.

Onix یک پایگاه‌داده تراکنشی پایدار پیاده سازی می‌کند که توسط یک ماشین وضعیت تکرار شونده، برای پخش کردن تمام به روزرسانی‌های وضعیت که نیازمند پایداری می‌باشد، به کار می‌رود و مدیریت سازگاری را ساده کرده است. پایگاه داده کپی شده با محدودیت‌های اجرایی زیادی روبه‌رو است، بنابراین تنها به عنوان یک مکانیزم ارسال معتبر برای تغییرات آهسته وضعیت شبکه به کار گرفته می‌شود. در صورت نیاز، پایگاه داده تراکنشی برای اپلیکیشن‌ها یک API پرس و جوی انعطاف پذیر مبتنی بر SQL به همراه مدل‌های ارزشمند داده جهت استفاده مستقیم فراهم می‌کند.

برای ترکیب پایگاه‌داده کپی شده باNIB ، Onixشامل ماژول‌های Import/Export می‌شود که با پایگاه‌داده در تعامل است. این بخش‌ها اعلان‌ها و ویژگی‌های خود را از پایگاه داده تراکنشی بارگیری و در آن‌ها ذخیره می‌کنند. اپلیکیشن‌ها می‌توانند به راحتی تعدیلات NIB را در یک تراکنش واحد گروه‌بندی کنند تا به پایگاه‌داده صادر شوند. زمانی که ماژول Import درخواست آغاز اعمال تغییر روی پایگاه داده را از پایگاه داده دریافت می‌کند، تغییرات بر NIB اعمال می‌شود.

در حالت کلی در Onix اطلاعات NIB را به روش می‌توان ذخیره کرد:

1. پایگاه داده :SQL پرهزینه است اما در عوض ماندگاری بالا دارد و مدیریت آن سخت‌تر است چون سرویس جداگانه‌ای برای پرس و جو از پایگاه داده می‌خواهد که سرعت را کاهش می‌دهد.

2. مدل مبتنی بر :DHT سریع‌تر است و

اطلاعات در دسترس تر هستند ولی ماندگاری در پایگاه داده را ممکن است نداشته باشد.

API هر دو مورد فوق در Onix پیش بینی شده است و بسته به انتخاب طراح و نیاز برنامه از هر کدام می‌خواهد می‌تواند استفاده کند.

معمولاً ایده کلی این است که اطلاعات توپولوژی و ساختار کلی شبکه و ویژگی‌های آن‌ها در یک پایگاه داده با ماندگاری بالا (SQL) ذخیره شود. اطلاعاتی که داخل مسیریاب‌هاست مثل جداول forwardingو چیزهایی که حالت موقتی دارند و زیاد تغییر می‌کنند در DHT ذخیره شود.

۹،۳- مدیریت وضعیت عناصر شبکه

طراحی Onix یک پروتکل خاص را برای مدیریت وضعیت ارسال عناصر شبکه تحمیل نمی‌کند. برعکس، نخستین واسط برای اپلیکیشن NIBاست و هر پروتکل مناسبی که عناصر شبکه پشتیبانی کنند، می‌تواند برای هماهنگ نگه‌داشتن NIB مستقل با وضعیت واقعی شبکه استفاده گردد. برای مثال OpenFlow یکی از پروتکل‌های مدیریت وضعیت عناصر شبکه است که Onix آن را حمایت می‌کند و در بخش‌های قبل به توضیح داده شد.

۹،۴- سازگاری و هماهنگی

یک NIB مرکز تعامل منابع داده‌ای متعددی است) نمونه‌های دیگر Onix و نیز عناصر شبکه .(مکانیزم‌های توزیع وضعیت به طور مستقیم با یکدیگر تعامل ندارند بلکه آن‌ها وضعیت را بهNIB ، Importیا Export می‌کنند. برای پشتیبانی اپلیکیشن‌های متعدد با نیازهای مقیاس پذیری و قابلیت اطمینان احتمالاً خیلی متفاوت، Onix نیاز به اپلیکیشن هایی دارد تا مشخص کنند که چه داده‌ای به یک منبع خاص وارد یا از آن خارج شود. اپلیکیشن ها این کار را از طریق پیکربندی ماژول‌های Import و Export انجام می‌دهند.

NIBمنابع داده را بدون نیاز به سازگاری زیاد ادغام می‌کند و در نتیجه وضعیت به روزرسانی می‌شود تا به NIB اضافه شود. در اینجا ممکن است به علت ناهماهنگی، وضعیت در هر منبع داده (DHT)یا به دلیل ناهماهنگی بین منابع داده‌ها تناقض ایجاد شود. بر این اساس Onix اپلیکیشن‌هایی را پیش بینی می‌کند که منطق تفکیک ناهماهنگی را در سیستم عامل نشان می‌دهند.

۱۰-شبیه سازMinninet

شبیه ساز Mininet شبکه‌ای مجازی همراه با کرنل‌های واقعی و سودو کدهای برنامه است و می‌تواند سوییچ‌های شبکه را بر روی یک ماشین (ماشین مجازی، ابر یا سیستم واقعی) به اجرا درآورد. این نرم‌افزار به دلیل امکان تعامل پذیری توسط خط فرمان و یا API می‌تواند در راستای توسعه، آموزش و تحقیق استفاده شود. همچنین این نرم‌افزار قابلیت‌های گسترده‌ای برای تست، ارزیابی و توسعه ابزارهای مبتنی بر شبکه‌های SDN و OpenFlow دارد.

۱۱-کارهای مرتبط

همان طور که بیان شد، Onixکار سیستم را کم می‌کند و در آن صفحه کنترل از صفحه داده مجزاست، ولی تمرکز Onix بر تبدیل شدن به یک پایگاه کنترل تولید – کیفیت برای شبکه‌های با مقیاس بالا، ما را قادر می‌سازد تا نسبت به سیستم‌های قبلی، بیشتر بر روی قابلیت اطمینان، مقیاس پذیری و عمومیت تمرکز کنیم. این سیستم اولین سیستمی نیست که کنترل شبکه را به عنوان یک مشکل سیستم توزیع شده درنظر می‌گیرد.

تحقیقات زیادی انجام شده که تمرکزشان بر ارائه یک صفحه ارسال توسعه پذیر به توسعه دهندگان شبکه بوده است) مانندRoutebricks ، Click، Onix .(XOPRدر ارائه یک صفحه کنترل توسعه پذیر، مکمل این سیستم‌ها محسوب می‌شود. به طور مشابه، Onixمی‌تواند برای ساختارهای انعطاف پذیر مرکز داده شبکه، مثل SEATTLE ، VL2و Portland پایگاهی برای مدیریت مراکز داده بزرگ باشد.

دیگر کارهایی که اخیراً انجام‌شده‌اند با توزیع وضعیت شبکه بین سوئیچ‌ها، از بار موجود در یک کنترلر متمرکز می‌کاهند Onix .به جای اینکه بر روی پیدا کردن یک رویکرد خاص در مقیاس مشخص تمرکز کند، بر روی مسئله فراهم کردن APIهای کلی مدیریت وضعیت توزیع شده متمرکز است .

Onixهمچنین راه بسیاری از سیستم‌های توزیع شده قبل از خود را در پیش گرفته است. سیستم‌هایی که برای پاسخ به الزامات پایداری به کمک اپلیکیشن ها متکی هستند تا بدین وسیله کارایی تکرار وضعیت را بهبود بخشندBayou .. ،PRACTI ،WheelFSو PNUTS نمونه‌هایی از این سیستم‌ها می‌باشند.

12-نتیجه گیری

SDNیک محصول یا مفهوم سخت‌افزاری/نرم‌افزاری نیست بلکه یک معماری و رویکرد جدید برای انعطاف پذیری و کنترل پذیری بیشتر شبکه‌ها و ظرفیت سازی برای استفاده از انواع برنامه‌های کاربردی، سرویس‌ها و خدمات نرم‌افزاری روی شبکه‌های کنونی است. از وظایف و نقش‌های پررنگ سخت‌افزار و تجهیزات شبکه می‌کاهد و به وظایف و نقش‌های لایه‌های نرم‌افزاری شبکه می‌افزاید و مدیریت و کنترل شبکه را ساده‌تر می‌کند .

الگوی SDN از پایگاه کنترل برای ساده سازی کاربردهای کنترل شبکه استفاده می‌کند. پایگاه کنترل به جای اینکه توسعه دهندگان را مجبور کند مستقیماً با جزئیات زیرساخت مواجه شوند، مسائل سطح پایین‌تر را خود رفع و رجوع می‌کند و توسعه دهندگان را قادر می‌سازد تا منطق کنترلشان را در یک API سطح بالا برنامه ریزی کنند. در این صورت، Onixمسائل شبکه را به مسائل سیستم‌های توزیعی تبدیل می‌کند تا با مفاهیم و الگوهای آشنا برای توسعه دهندگان سیستم‌های توزیعی قابل حل باشد.

لازم به ذکر است که Onix به تنهایی نمی‌تواند تمامی مشکلات مدیریت شبکه را حل کند. هنوز هم لازم است طراحان اپلیکیشن های مدیریتی کاربردهای مقیاس پذیری طراحی خود را درک کنند .Onix ابزارهایی عمومی برای مدیریت وضعیت فراهم می¬کند، ولی نمی‌تواند تمام مسائل و مشکلات مربوط به مقیاس پذیری و سازگاری را هموار سازد. ما هنوز درباره ساختن منطق کنترل در APIاونیکس در حال یادگیری هستیم، ولی در نمونه‌هایی که تا به حال با آن‌ها روبرو شده‌ایم ساخت اپلیکیشن های مدیریت با Onix بسیار ساده‌تر از ساخت آن‌ها بدون Onix است .

چکیده

یکی از جدیدترین فناوری‌هایی که کاربرد پرداخت آن نیز به سرعت در حال گسترش است، فناوری RFID است. سیستم اصلی RFID شامل سه قسمت است: یک آنتن، یک فرستنده و گیرنده و یک .RF آنتن یک پل بین برچسب، و فرستنده و گیرنده‌است RFID .در محدوده‌های گوناگون فرکانسی شامل ۱۲۵KHz، ۱۳،۵۶MHz، ۲،۴۵GHz، ۵،۸GHz و ۸۶۰-۹۵۰ MHz کار می‌کند. هم‌چنین محدوده‌های خواندن را از چند سانتی متر تا ۵ متر ارائه می‌کند که به فرکانس دستگاه‌ها وابسته‌است. فناوری RFID از امواج رایویی به منظور شناسایی اشیاء استفاده می‌کند. یک سامانه عبارت است از یک برچسب حاوی یک آنتن و یک تراشه به همراه اطلاعاتی در مورد کالا یا اشیاء و هم‌چنین یک قرائت‌گر که دارای یک فرستنده و گیرنده رادیویی می‌باشد. کاربردهای اولیه این فناوری در بهبود کارایی سامانه های اطلاعاتی زنجیره‌های تامین و توسعه فروشگاه‌های بزرگ زنجیره‌ای بوده‌است. لکن کاربردهای این فناوری در حال گسترش بوده، به طوری که در ردیابی کالا، افراد، و حفاظت از مناطق امن از RFID استفاده شده‌است. نگهداری تراکنش‌ها و خدمات پرداخت فناوری RFID نیز از جمله کاربردهای آن می‌باشد که در خریدهای خرد روزانه شهروندان به کار گرفته شده‌است.

در این مقاله قصد داریم که با فناوری ,RFIDمزایا ، معایب ، کاربرد ها و چالش های آن نظیر انواع حملات و راه های مقابله با این چالش ها بپردازیم.

۱-۱مقدمه

امروزه ضرورت شناسایی خودکار عناصر و جمع آوری داده مرتبط به آنان بدون نیاز به دخالت انسان جهت ورود اطلاعات در بسیاری از عرصه های صنعتی ، علمی ، خدماتی و اجتماعی احساس می شود . در پاسخ به این نیاز تاکنون فناوری های متعددی طراحی و پیاده سازی شده است . به مجموعه ای از فناوری ها که از آنان برای شناسایی اشیاء ، انسان و حیوانات توسط ماشین استفاده می گردد ، شناسایی خودکار و یا به اختصار Auto ID گفته می شود . هدف اکثر سیستم های شناسایی خودکار ، افزایش کارآیی ، کاهش خطاء ورود اطلاعات و آزاد سازی زمان کارکنان برای انجام کارهای مهمتر نظیر سرویس دهی بهتر به مشتریان است .

تاکنون فناوری های مختلفی به منظور شناسایی خودکار طراحی و پیاده سازی شده است . کدهای میله ای ، کارت های هوشمند ، تشخیص صدا ، برخی فناوری های بیومتریک ، OCR (برگرفته شده از optical character recognition) و RFID نمونه هایی در این زمینه می باشند .

در حال حاضر شرکت های بزرگی مثل مک دونالد و وال مارت از این فناوری استفاده می کنند. در ادامه با فناوری RFID بیشتر آشنا خواهیم شد .

۲-۱ RFID چیست؟

RFIDسامانه شناسایی امواج رادیویی (به انگلیسی: Radio Frequency Identification) (به اختصار RFID) سامانه شناسایی بی سیمی است که قادر به تبادل داده‌ها به وسیلهی برقراری اطلاعات بین یک Tag که به یک کالا، کارت و... متّصل شده و یک بازخوان (Reader) است. سامانه های RFID از سیگنال‌های الکترونیکی و الکترومغناطیسی برای خواندن و نوشتن داده‌ها بدون تماس بهره می برند .

۲-۲ آشنایی با فناوری RFID

تصوّر کنید که وارد یک فروشگاه زنجیره‌ای شده‌اید و اقلام مورد نیاز خود را داخل چرخ دستی قرار داده اید. صندوق دار با استفاده از بار کد باید تک تک اقلام داخل سبد را برداشته و اطلاعات آن را توسط بارکد خوان یکی یکی به داخل کامپیوتر وارد کند تا فاکتور اقلام انتخابی شما صادر گردد. بسیاری از اوقات به دلیل آنکه تعداد کالاهای خریداری شده بسیار زیاد می‌باشند؛ صف های طولانی ای در فروشگاه های زنجیره ای مشاهده می شود. گاهی اوقات نیز مخدوش شدن علائم بار کد، از خواندن اطلاعات جلوگیری می‌کند، که این خود موجب مشکلات بیشتری می‌شود. اما با این فناوری جدید یعنی RFID شما سبد کالای خود را برمی دارید و بدون اینکه مجبور به ایستادن در صف های طولانی شوید و یا حتی بدون اینکه مجبور باشید اقلام خریداری شده را به صندوقدار یا نگهبان نشان دهید، از در خارج می‌شوید.

حتما می پرسید چرا؟ چون شناسه روی کالا دیگر بارکد نیست بلکه از نوع RFID می‌باشد و خودش با فرستان علائم رادیویی کلیه اطلاعات جاری خود از قبیل تعداد، قیمت، وزن، ... را به کامپیوترهای موجود در درب های خروجی مخابره و منتقل می‌کند.

۳- اجزای یک سیستم RFID

۳-۱ شناسه یا Tag

۳-۲ بازخوان بر چسب یاReader

۳-۳ آنتن

۳-۴ نرم‌افزار مدیریت اطلاعات

۳-۱ Tag یا شناسه : این شناسه‌ها دارای دو بخش تراشه و آنتن هستند و دارای عملکرد بسیار ساده‌ای می‌باشند؛ تراشه اطلاعات را از طریق آنتن منتشر می‌کند و حسگرهایی که در اطراف قرار دارند، این اطلاعات را دریافت می‌کنند.

تقسیم بندی انواع tag ها

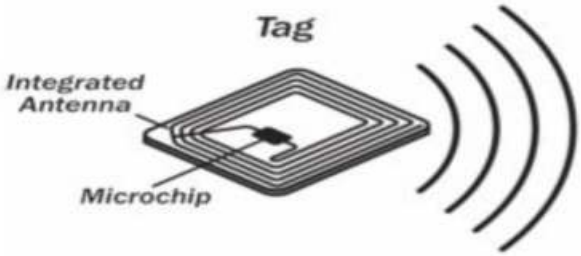
۱-۱-۳دسته بندی اصلی

Tag های غیر فعّال، باتری ندارند و انرژی لازم برای فعّال سازی پردازنده و ارسال اطلاعات خود را از سیگنالی که از آنتن Reader دریافت می کنند، تأمین می‌کنند؛ بدین ترتیب که ابتدا Reader امواجی را به سمت Tag ارسال می کند، سپس پردازنده Tag با استفاده از انرژی این امواج، بیدار شده و دستور دریافتی از سوی Reader را پردازش می‌کند و توسط امواجی که انرژی آن ها هم از امواج دریافتی تامین شده است، نتیجه را به Reader مخابره می‌کند. بدین ترتیب واضح است که برد مسافتی این Tag ها بسیار محدود است؛ چرا که فاصله Tag از آنتن باید به اندازه ای باشد تا انرژیی که Tag دریافت می‌کند، توان بیدار کردن پردازنده و تامین انرژی موج برگشتی را داشته باشد. معمولا برد Tag های غیر فعال از ۵ سانتی متر تا ۵ متر است.

از طرف دیگر Tag های فعال، انرژی لازم برای پردازنده و ارسال امواج به سمت آنتن را از باتری همراه خود تامین می‌کنند. بدین ترتیب برد این Tag ها در برخی موارد تا چند صد متر هم می‌رسد. قیمت این Tag ها در حدود چند ده دلار است که به مراتب از Tag های غیر فعال (با قیمتی در حدود چند ده سنت) گران قیمت تر هستند. یک پارامتر مهم در انتخاب Tag های اکتیو یا همون فعال، طول عمر باتری آن است، چرا که پس از اتمام باتری، شما مجبورید باتری یا اغلب ، خود Tag را تعویض کنید و این باعث افزایش هزینه ها می‌گردد. بدین ترتیب، طول عمر باتری یک عامل تعیین کننده در هزینه نهایی سیستم شما خواهد بود. Tag های RFID هر کدام دارای یک کد منحصر به فرد می‌باشند به طوریکه هیچ دو برچسبی در دنیا تولید نخواهد شد که کد یا ID یکسانی داشته باشند و کلیه تولید کنندگان Tag تحت نظر انجمن بین المللی مبادرت به ثبت آنها می‌نمایند. این Tag ها از دو بخش اصلی تشکیل شده‌اند که بخش اول یک آنتن مایکرواستریپ بوده و بخش دوم یک چیپ الکترونیکی می‌باشد. لازم به ذکر است که این بخش‌ها به صورت کاملا مسطح و بسیار کوچک هستند به طوریکه می‌توانند آنها را در درون کاغذ یک برچسب معمولی قرار دهند و از دید کاربر، تنها یک برچسب معمولی کاغذی خواهد بود. در بخش چیپ الکترونیکی Tag ها یک فرستنده و یک گیرنده امواج رادیویی به همراه مدارهای پردازشگر الکترونیکی قرار دارد. در برخی از انواع این چیپها، حافظه نیز به مقدار دلخواه وجود دارد به طوریکه هرنوع اطلاعات مربوط به کنترل و دسترسی می‌تواند مستقیما روی برچسب و درحافظهی آن نوشته شود. در این صورت این برچسب درست مانند یک شناسنامهی الکترونیکی همراه محصول عمل خواهد کرد.

۲-۳ دسته بندی با در نظر گرفتن منبع انرژی تامین کننده

در صورتی که بخواهیم Tag ها را با در نظر گرفتن منبع انرژی تامین کننده شان دسته بندی کنیم به ۴ دسته اصلی تقسیم بندی می‌شوند:



شکل۱- یک شناسه rfid

۳-۲-۱- Tagهای غیر فعال Passive Tags

که انرژی و برق مورد نیاز خود را از Reader ها به وسیله یکسری از روش‌های تراگسیل بدست می‌آورند.

۳-۲-۲- Tagهای فعال Active Tags

که انرژی مورد نیازشان توسط یک باتری داخلی و جهت برقراری ارتباط دارای یک پردازنده، یک حافظه و حسگر می‌باشند.

۳-۲-۳- Tag های نیمه غیر فعال Semi-Passive Tags

که علاوه بر استفاده از باتری داخلی شان، می‌توانند از انرژی منتقل شده توسط بازخوان بر چسب Reader ها نیزبهره مند شوند.

۳-۲-۴ Tagهای دو طرفه Two way Tags

که علاوه براستفاده از باتری داخلی شان می‌توانند بدون کمک گرفتن از Reader ها دیگر اقسام هم شکل خود را نیز شناسایی کرده و با آن‌ها به گفتگو بپردازند.

۵ -چالش ها

۱-۵سازگاری نرم افزاری

سازگاری اطلاعات موجود بر روی برچسب هایRFID با نرم‌افزارهای کاربردی موجود، یکی از جنبه‌های مهم معرفی فناوریRFID است. اطلاعات جمع‌آوری شده توسط RFID reader ها نیاز به تولید ارزش دارد. مثلاً وضوح بیشتر زنجیره عرضه و یا برنامه ریزی بهینه تر. این موضوع که بکارگیریRFID در ابعادی بسیار بزرگ در زنجیره تولید و عرضه ممکن است به انفجار اطلاعات جمع‌آوری شده منجر شود، امری بسیار حائز اهمیت است و این امر نیاز به نرم‌افزاری را که فاصله خالی بین اطلاعات دریافتی درRFID reader ها و اطلاعات موجود بر روی نرم‌افزارها را پرکند، به وجود می‌آورد؛ بنابراین جای تعجب نیست که شرکت‌های تولید کننده نرم‌افزارهای رابط (میان فرستنده‌ها و گیرنده‌ها)، موسسات برنامه ریزی ذخایر (ERP) و فروشندگان سیستم‌های مدیریت انبارداری، برنامه‌های خود را برای همکاری در زمینه تولیدRFID اعلام کرده‌اند.

۲-۵ رخنه گری

اگر به پیشینه مقوله رخنه‌گری نگاهی بیندازید، خواهید دید با آمدن هر فناوری جدید، رخنه‌گران به دنبال رخنه‌های امنیتی آن گشته و با سوء استفاده از حفره های موجود در آن، فناوری را به چالش می‌کشند. رخنه‌گران با دسترسی و کنترل سیستم می‌توانند دست به اقدامات خرابکارانه زنند و باعث شوند هزینه ای هنگفت از حیث مادی و معنوی و همچنین خسارت‌های جبران ناپذیری به فرد و سیستم تحمیل شود. بدافزارها_ نرم‌افزارهای مخربی هستند که هدف اصلی‌شان تخریب سیستم‌های کامپیوتری می‌باشد. همان‌طور که انتظار آن می‌رفت، بدافزارها به سیستم‌های مبتنی بر RFID نیز راه یافته و آن‌ها را به چالش کشیدند. اهمیت این مقوله و این‌که فناوری RFID از بدافزارها در امان نیست، ما را بر آن داشت که به بررسی تاثیر بدافزارها و چالش‌های امنیتی ایجاد شده توسط آن‌ها بپردازیم. با بررسی صورت گرفته سه نوع اصلی بدافزار در سیستم های مبتنی بر RFID شناسایی شده است. این سه نوع استثمارگرها، کرم‌ها و ویروس‌ها می‌باشند. بدافزارها منشاء خارجی داشته و بر اساس برچسب های RFID به وجود می‌آیند. در نهایت، تاثیر بدافزارها و چالش‌های امنیتی ایجاد شده توسط آن‌ها را ارائه داده‌ایم.

RFID امروزه در بسیاری از صنایع دنیا استفاده می شود. به عنوان مثال، در فروشگاه‌ها از RFID به جای سیستم قدیمی بارکد (رمزینه) استفاده می‌شود. نمونه دیگر در صنعت مسافربری، گذرنامه های مسافران می‌باشد. همان‌طور که می‌دانید گذرنامه ها حاوی اطلاعات ارزشمند شخصی هستند و اگر یک رخنه‌گر بتواند رخنه ای در سیستم RFID گذرنامه بیابد، سوء استفاده‌های فراوانی از آن شخص و به نام او می‌تواند انجام دهد. با این‌که داده‌های موجود در گذرنامه ها رمزنگاری می‌شوند ولی باز هم سوء استفاده هایی در این زمینه دیده شده است. رخنه‌گری و سرقت اطلاعات از همان روزی شروع شد که هیچ رایانه ای وجود نداشت. با پا به عرصه گذاشتن رایانه و شبکه، **رخنه‌گری** و سرقت اطلاعات شیوه علمی به خود گرفت و روز به روز متکامل‌تر شد. کلا چهار نوع حمله می توانیم داشته باشیم:

- حمله از نوع وقفه_: بدین معنا که حمله کننده باعث شود شبکه مختل شده و مبادله اطلاعات امکان پذیر نباشد.
- حمله از نوع استراق سمع_: بدین معنا که حمله کننده به نحوی توانسته است اطلاعات در حال تبادل روی شبکه را گوش داده و بهره برداری نماید
- حمله از نوع دستکاری داده ها_: یعنی حمله کننده توانسته است به نحوی اطلاعاتی که روی شبکه مبادله می شوند را تغییر دهد. بدین صورت که داده های دریافتی در مقصد، متفاوت با آن چیزی باشد که از مبدا، ارسال شده است.
- حمله از نوع افزودن اطلاعات: یعنی حمله کننده اطلاعاتی را که در حال تبادل روی شبکه است، تغییر نمی دهد بلکه اطلاعات دیگری را که می تواند مخرب یا بنیان‌گذار حملات بعدی باشد، به اطلاعات اضافه می‌کند.

بر اساس این چهار نوع حمله ای که معرفی شد، حملات زیادی با نام‌های گوناگون به وجود آمده است. تحقیقات فراوانی در مورد RFID و امنیت آن صورت گرفته است که هر یک وارد بحث تفصیلی در این مقوله شده است.

فلدهوفر و همکارانش به بررسی الگوریتم رمزنگاری AES در حوزه RFID پرداخته اند. گروس‌شادل و تیلیچ، RFID بر مبنای کلید عمومی و خصوصی را بررسی کرده اند.

جولز و همکارانش، بحثی در مورد برچسب‌های RFID_ دارند و راهکارهای افزایش امنیتشان را بررسی کرده اند. کارجوت و مسکوویتز، برچسبی را پیشنهاد داده اند که امنیت بالاتری نسبت به نسل قبلی خود دارد. لینگل، روش جدیدی برای پویش کردن حوزه‌های RFID ارائه داده است. بورک، به مساله تراشه های RFID_ و امنیت آن‌ها می‌پردازد.

هدف ما در این مقاله این است که در ابتدا، بدافزارهای موجود در سیستم‌های مبتنی بر RFID را معرفی کرده و در نهایت چالش‌های امنیتی ایجاد شده توسط آن‌ها را ارائه دهیم.

دسته بندی بدافزارهای RFID

بدافزارها، نرم افزارهای مخربی هستند که هدف اصلی‌شان تخریب سیستم‌های کامپیوتری می‌باشد. بدافزار در RFID، از طریق برچسب RFID منتقل شده و اجرا می‌شود. در این قسمت بدافزارهای رایج RFID معرفی می‌شوند.

۵-۲-۱ استثمارگر RFID_

استثمارگر RFID، یک داده برچسب RFID می‌باشد که قسمتی از سیستم RFID را که با آن مواجه است، مورد بهره برداری قرار می‌دهد. وقتی قرائت‌گرRFID_ ، یک برچسب را پویش_ می‌کند، انتظار دارد که اطلاعات را در قالب معینی دریافت کند. رخنه‌گر می‌تواند داده‌هایی تولید کند که قالب و محتویات آن با قالب مورد انتظار دستگاه قرائت‌گر یکسان نباشد و این باعث می‌شود که نرم افزار RFID قرائت‌گر و حتی پایگاه داده آن را تخریب کند و یک حمله از نوع وقفه را شکل دهد.

۵-۲-۲ کرم‌های RFID_

کرم برنامه‌ای است که در طول شبکه انتشار می‌یابد و در طول انتشار خود، نقص‌های امنیتی را در سرویس‌هایی که به طور گسترده استفاده می شوند پیدا می‌کند. کرم از ویروس قابل شناسایی می باشد چرا که کرم‌ها نیازی به فعالیت کاربر برای انتشار ندارند. کرم‌های RFID از اتصالات شبکه، سوء استفاده می‌کنند تا خودشان را تکثیر کنند. این کرم ها توسط سوء استفاده از سرویس های برخط_در RFID یا حتی از طریق برچسب‌های RFID ، منتشر می شوند. کدهای موجود در کرم‌ها، موجب می شوند تا کارسازهای RFID ، برخی از فایل‌ها را از مکان‌های دور، بارگیری_ کرده و سپس آن‌ها را اجرا کنند. این فایل‌ها در واقع کارساز میان‌افزار_می‌باشند. کرمی که نرم افزارهای RFID را آلوده کرده است می تواند برچسب‌های RFID را نیز آلوده کند.

۵-۲-۳ ویروس‌های RFID_

یک ویروس RFID ، به‌صورت خودکار،کدش را در برچسب‌های RFID ، بدون نیاز به اتصالات شبکه تکثیر می‌کند. ویروس‌های RFID خواه ناخواه یک بار مفید_دارند که عملکرد سیستم پشتیبان RFID را تغییر می‌دهند یا تخریب می‌کنند.

از آنجایی که برچسب‌های جدید آلوده شده RFID در مسیر خودشان ادامه حرکت می‌دهند، آن‌ها می‌توانند سیستم‌های دیگر RFID را آلوده کنند.

۵-۲-۱-۱چالش‌های امنیتی ایجاد شده توسط بدافزارها

در قسمت قبل بدافزارهای موجود در سیستم‌های مبتنی بر RFID را معرفی اجمالی کردیم. در این قسمت به شرح چالش‌های امنیتی ایجاد شده توسط آن‌ها می‌پردازیم.

۵-۲-۲-۱چالش‌های امنیتی ایجاد شده توسط کرم‌ها

کرم‌ها معمولا دارای بار مفید هستند که فعالیت‌ها را از حذف فایل‌ها تا ارسال اطلاعات، از طریق پست الکترونیکی و نصب نرم افزارها، انجام می‌دهند. یکی از متداول‌ترین بار مفیدها برای یک کرم، عبارتست از نصب درب پشتی_در رایانه آسیب دیده که برگشت آسان رخنه‌گر را به سیستم آن رایانه، در آینده امکان پذیر می‌سازد. یک کرم RFID ، از طریق پیدا کردن نقص‌های امنیتی در خدمات RFID اینترنتی، انتشار پیدا می‌کند. کرم‌های RFID ، لزوما نیازی به فعالیت کاربران، برای انتشار ندارند و آن‌ها به راحتی می‌توانند از طریق برچسب‌های RFID منتشر شوند. برچسب‌های RFID معمولا برای در بر گرفتن کل کرم، بسیار کوچک می‌باشند. بنابراین برچسب تنها، تعداد کافی از کرم‌ها را در بر خواهد گرفت که می‌تواند بقیه آن‌ها را از رایانه متصل به اینترنت بارگیری کند . برچسب RFID ممکن است شامل کدهای درختی_برای بارگذاری باشند. اجرای کرم ها و یا دستور برنامه واسط، معمولا به فضای کمتری از کدهای درختی نیاز دارند و همچنین از قابلیت حمل بیشتری برخوردار هستند. با این وجود، ممکن است برخی نقص‌ها تنها امکان اجرای کدهای درختی را فراهم کنند. کرم‌ها ممکن است هر بخش از میان افزار RFID را برای انتشار مورد هدف قرار دهند. برخی سیستم‌های پایگاه داده، دستورات SQL را ارائه می‌دهند که دستورات برنامه واسط را روی کارساز پایگاه داده اجرا خواهد کرد. این دستورها ممکن است برای بارگیری و اجرای کرم‌ها، مورد سوء استفاده قرار گیرند. هر بخش میان افزار که به زبان C++ و یا C نوشته شده باشد، ممکن است در معرض سرریز میانگیر_قرار گیرد که می‌تواند برای وارد کردن کد درختی که انتشار را انجام می دهد، به کار گرفته شود. ما با استفاده از سرریز میانگیر، هیچ کرمی را توسعه ندادیم، اما نمونه سرریز میانگیر که دستور برنامه واسط را اجرا می‌کند، ممکن است برای ایجاد یک کرم استفاده شود. این کار از طریق اجرای یکی از دستورات برنامه واسط قبلی، انجام می‌پذیرد.

۵-۲-۳-۱چالش‌های امنیتی ایجاد شده توسط ویروس‌ها

یک ویروس دو کار اصلی انجام می‌دهد: خودش را تکثیر می‌کند و به طور گزینشی_بار مفید اجرا می‌کند. ویروس RFID برای تکثیر خودش، از پایگاه داده استفاده می‌کند. جزئیات تکثیر، به پایگاه داده استفاده شده بستگی دارد؛ اما دو گروه از ویروس ها قابل تشخیص می‌باشند. گروهی که از پرسش‌های خود ارجاعی_استفاده می‌کنند و گروه دیگری که از قابلیت خودتکثیری_ استفاده می‌کنند.

بارمفیدی که ویروس می تواند اجرا کند به مکانیسم خود تکثیر و پایگاه داده مورد هدف بستگی دارد.

۵-۲-۴تکثیر با استفاده از پرسش های خود ارجاعی

سیستم های پایگاه داده معمولا شیوه ای برای به دست آوردن پرسش‌های در حال اجرا برای هدف نظارت سیستم ارائه می‌دهند. با این وجود، این عملکردها، پرسشی را به صورت یک ردیف طبیعی بر می‌گرداند که امکان ذخیره آن ها در پایگاه داده را فراهم می‌کند . ما دو نوع ویروس را گسترش داده‌ایم: نوعی که در یک پرسش واحد وجود دارد و نوعی که به پرسش های چند گانه نیاز دارد. ویروس به کار گیرنده پرسش واحد به مشخصات کمتری از پایگاه داده نیاز دارد، اما نمی‌تواند کد SQL را به عنوان یک بار مفید حمل کند. ویروس استفاده کننده از پرسش چندگانه نیازمند پایگاه داده می‌باشد که از آن پشتیبانی کند، اما این ویروس از امکان حمل کد SQL برخوردار می باشد .

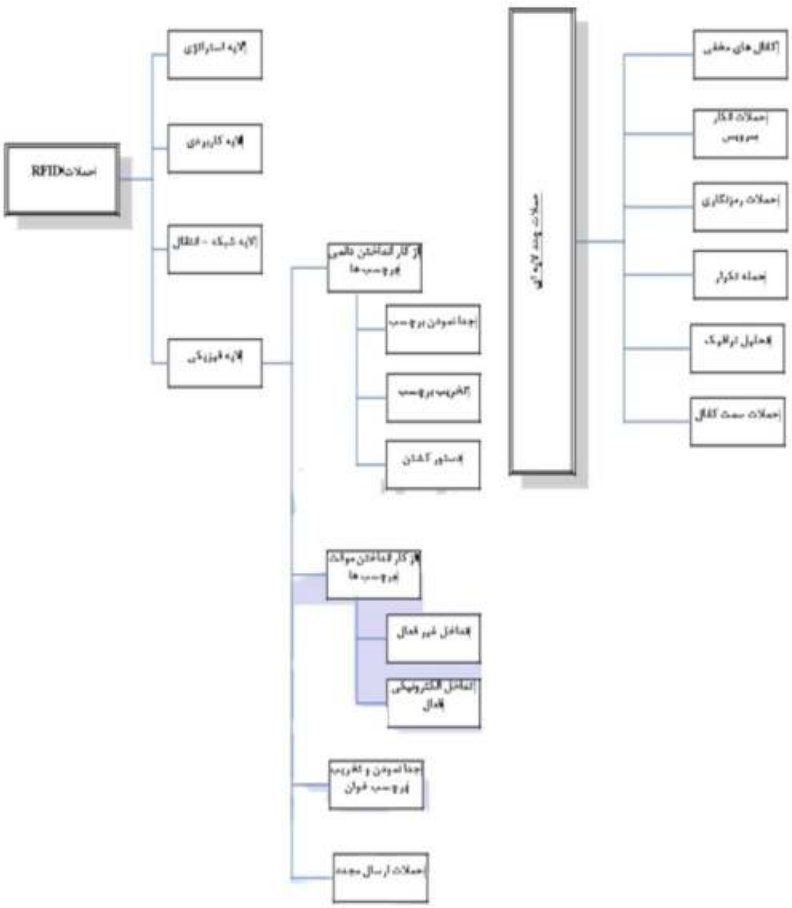
۵-۲-۵ تکثیر با استفاده از قابلیت خودتکثیری

خودتکثیر، برنامه‌ای است که کد منبعش را چاپ می‌کند . ویروس با کپی کردن کد منبعش در پایگاه داده، می‌تواند خودش را تکثیر کند. خودتکثیر نیازمند پرسش‌های چندگانه می‌باشد و به این معناست که آن‌ها در تمامی پایگاه‌های داده، پشتیبانی نمی‌شوند. با این وجود، آن‌ها امکان اجرای کد SQL را به صورت یک بارِ مفید فراهم می‌کنند.

۵-۲-۶ بارمفید

بسته به نوع ویروس و پایگاه داده مورد استفاده، انواع مختلفی از بارمفید وجود دارد. در صورت استفاده ویروس، از پرسش های چندگانه، می‌توان کد SQL را روی پایگاه داده اجرا کرد. بسته به اجازه پایگاه داده، ممکن است پایگاه داده حذف شود. در صورت استفاده از واسط مدیریت بر مبنای وب، ممکن است اجرای نسخه جاوا در جویشگر کاربر واسط مدیریت، میسر شود . در شرایط بد ممکن است دستور برنامه واسط روی کارساز وب و یا کارساز پایگاه داده اجرا شود. در این صورت، آسیب تنها به‌واسطه اجازه‌ای که دستورها با آن اجرا می‌شوند محدود خواهد شد.

۶- انواع حملات



3D sensors

سنسور چیست؟

حسگر یا سنسور، المان حس کننده‌ای است که کمیت‌های فیزیکی مانند فشار، حرارت، رطوبت، دما و... را به کمیت‌های الکتریکی پیوسته (آنالوگ) یا غیرپیوسته (دیجیتال) تبدیل می‌کند. در واقع سنسور یک وسیله‌ی الکتریکی است که تغییرات فیزیکی یا شیمیایی را اندازه‌گیری و آن را به سیگنال الکتریکی تبدیل می‌کند. سنسورها می‌توانند عمق فاصله یا محدوده را اندازه‌گیری کنند.

سنسورها در انواع دستگاه‌های اندازه‌گیری، سیستم‌های کنترل مانندplc مورد استفاده قرار می‌گیرند. به طورکلی دو نوع سنسور وجود دارد:

- projected-light-sensor: که این نوع سنسور نور را با الگوی دوربین دوبعدی استاندارد ترکیب کرده و عمق را اندازه‌گیری می‌کند.
- Time-Of-Flight: سنسورهای زمان‌سنجی هستند که عمق را با تخمین زدن تأخیر زمانی اندازه‌گیری می‌کنند. (تاخیر زمانی از انتشار نور تا تشخیص نور).

سنسورهای سه‌بعدی خطوط ارتفاع را تعیین می‌کنند. داده‌های سه‌بعدی می‌توانند با حرکت دادن اشیا یا با حرکت سنسور ایجاد شوند. داده‌های سه‌بعدی اندازه‌گیری شده می‌توانند برای پردازش خارجی یا پردازش در سنسور برای اندازه‌گیری یا تشخیص مناسب، شبیه‌سازی شوند.

حسگرهای سه‌بعدی همچنان در مراحل اولیه‌ی توسعه قرار دارند. سنسورهای سه‌بعدی کاربرد وسیعی در زندگی مدرن دارند که در محدوده‌ی هواپیماهای بدون سرنشین، ربات‌ها؛ و به طور کلی برای دسترسی به زندگی واقعی استفاده می‌شوند.

ما می‌توانیم اقدامات خود را با حرکات ساده، حرکات بدن؛ و حتی تشخیص چهره، کنترل کنیم. لیزرهای پیشرفته به ویژه لیزرهای دیود و سنسورهای سه‌بعدی در حال تغییر، چگونگی تعامل با تکنولوژی هستند. واقعیت مجازی، واقعیت افزوده و وسایل رانندگی مستقل؛ تنها نمونه‌ای از برنامه‌های جدید در افق هستند. با گسترش سریع بازار سه‌بعدی و با افزایش نیاز به سنسورهای سه‌بعدی جای تعجب نیست که این سخت‌افزار در حال تبدیل شدن به قابلیت دسترسی بیشتر است.

حسگر سه‌بعدی را می‌توان با استفاده از تعدادی فناوری‌های مختلف به دست آورد. هر یک از این فناوری‌ها که مورد استفاده قرار می‌گیرند، دارای نقاط قوت جداگانه هستند

(Jabil optics) توانایی طراحی، توسعه و استفاده از سخت‌افزار حسگر سه‌بعدی بر اساس هر یک از سه تکنولوژی که در ادامه به توضیح آن خواهیم پرداخت، یا با استفاده از یک پلتفرم موجود یا طراحی ماژول به صورت جداگانه را دارد.

سه تکنولوژی فوق به شرح زیر است:

۱) **بینایی استریوسکوپی:** این تکنولوژی از عملکرد چشمان انسان برای دیدن تصاویر گرفته شده است. دو دوربین در موقعیت‌های کمی انداز (درست مثل چشم‌های انسان) قرار می‌گیرند. سپس دو تصویر گرفته شده به یک تصویر متحد می‌شوند. تفاوت‌های کوچک ناشی از موقعیت‌های مختلف دوربین باعث ایجاد تصویر کلیشه‌ای، یعنی تصویر سه‌بعدی می‌شود. (دوربین استریو) یک سنسور است که به دید استریو متکی است و یک نقشه تقریبی از اطلاعات عمق در آن ذخیره می‌شود. مزیت دید استریو آن است که قادر به کار کردن در خارج از منزل است، اما دوربین‌های فعلی هنوز بسیار دقیق نیستند. علاوه بر این داده‌های تصویری استریو نیاز به زمان زیادی برای پردازش دارند و الگوریتم‌ها معمولاً بسیار محاسباتی هستند. چشم انداز استریوسکوپی یک کمک اضافی از یک ماژول نقطه در شیء یا صحنه برای کمک به تمرکز دوربین است.

۲) **الگوی ساختار نور:** یک الگوی نور از هر دو خط، مربع (ساختارهای دوره‌ای) یا نقطه‌ها بر روی یک شی یا یک صحنه با یک ماژول پیش‌بینی می‌شود. یک دوربین مستطیلی ماژول طرح‌ریزی و سپس نور منعکس را ضبط می‌کند.

۳) **الگوی ایجاد شده توسط مثلث،** بین طرح‌ریزی و ماژول بین دو دوربین برای به دست آوردن جسم یا صحنه است.

۴) **زمان پرواز:** فلاش‌های مستقیم کوتاه یا فلاش‌های کوتاه سینوسی از طریق یک ماژول طرح ریزی منتشر می‌شوند و سپس توسط یک ماژول دوربین گرفته می‌شوند. زمان سفر نور از امیتر (emitter) به جسم و بازگشت به دوربین محاسبه می شود. سپس مختصات اندازه‌گیری شده یک نصویر سه‌بعدی ایجاد می‌کند.

هر تکنولوژی نقص‌هایی دارد. یکی از نقص‌های اصلی در ارتباط با تکنولوژی سنسورهای سه‌بعدی، میزان مصرف انرژی در هنگام مصرف و استفاده از آن است.

تولیدکنندگان سنسورهای تصویربرداری سه‌بعدی برای راه‌اندازی حسگرهای سه‌بعدی که دیدشان از دیوار می‌تواند نفوذ کند، تلاش می‌کنند. این ویژگی که هدف شرکت‌های کابلی، پهنای باند و بخش‌های هوشمند خانه است، راه طولانی را برای پیشرفت خواهد داشت.

شرکت‌های تلفن‌های هوشمند هم سعی دارند تا حسگرهای سه‌بعدی را در دستگاه‌های خود قرار دهند. به تازگی سامسونگ گلکسی نوت ۸ در نسخه ی خود دوربینی با تکنولوژی سه‌بعدی قرار داده است.

(تشخیص حرکت برای رابط کاربر نیز با حسگر سه‌بعدی انجام می‌شود. برخورد با یک منبع نور مادون قرمز در یک عنصر بصری در یک الگوی ساختار یافته یا یک ورق نور، باعث می‌شود که شما بتوانید بازی‌ها یا دستگاه‌های سرگرمی را با حرکات خود کنترل کنید.)

ما در جهان سه‌بعدی زندگی می‌کنیم که به معنای تجزیه و تحلیل رفتار انسان است. پس نیاز به وسایلی با درک اطلاعات سه‌بعدی است. با استفاده از سنسورهای سه‌بعدی می‌توان وضعیت واقعی پیام‌های محیطی را شناسایی کرد.

کاربرد سنسور سه‌بعدی در گوشی همراه اپل (apple)

سنسور سه‌بعدی مبتنی بر لیزر جدیدترین تکنولوژی موبایلی است که اپل مشغول کار بر روی آن است. این سنسور موبایل در سرویس‌ها و برنامه‌های واقعیت افزوده کاربرد خواهد داشت.

اپل بعد از مرفی آیفون ایکس، که مهم‌ترین و پیشرفته ترین سیستم آن سیستم تشخیص چهره (فیس آیدی)

است، اعلام کرد توسعه این گوشی موبایل چند سال زمان برده است، پس بعید نیست این شرکت هم‌اکنون نیز مشغول توسعه یک تکنولوژی نوین برای آیفون‌های سال‌های بعد باشد.

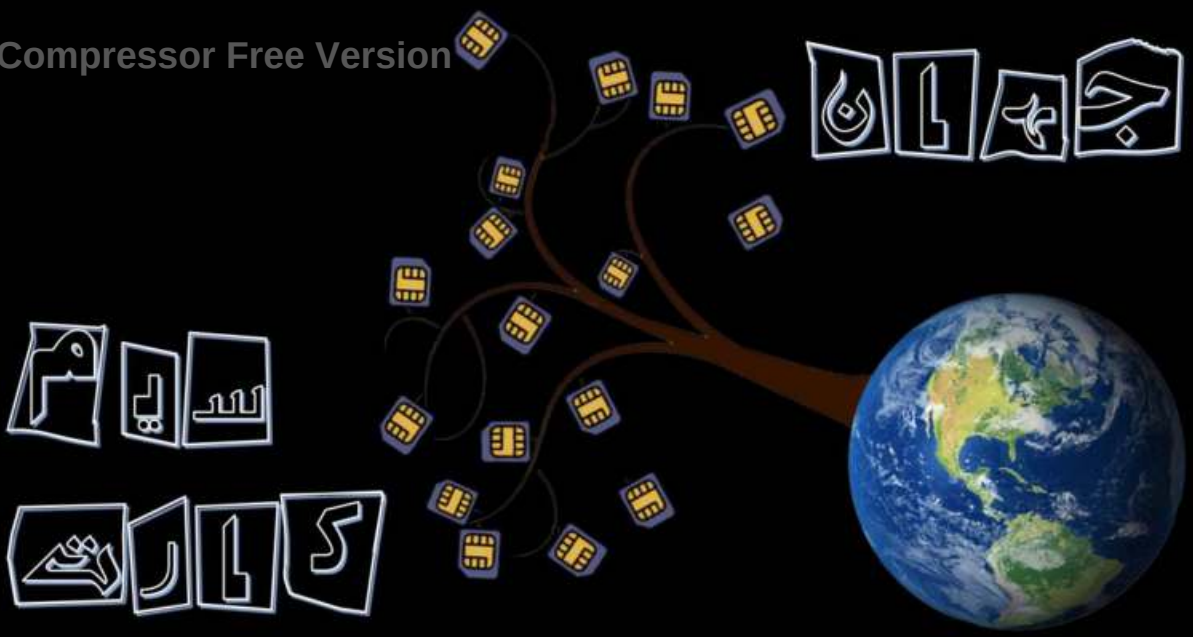
بر اساس جدیدترین گزارش Bloomberg اپل مشغول توسعه یک سنسور سه‌بعدی مبتنی بر لیزر است که به احتمال زیاد در برترین آیفون سال ۲۰۱۹ به کار خواهد رفت. کاربرد این سنسور در اپلیکیشین‌ها و سرویس‌های واقعیت افزوده خواهد بود که تکنولوژی آینده متعلق به آن بوده و اپل بر روی آن سرمایه‌گذاری زیادی کرده است.

این سیستم با منعکس کردن لیزر از دستگاه و سپس اندازه‌گیری زمانی که بازگشت اشعه به طول می‌انجامد، یک نقشه تشکیل خواهد داد. چنین اطلاعات تفصیلی و عمیقی را می‌توان با استفاده از دوربین‌های دوگانه نیز به دست آورد، چنان‌که اپل نیز در حال حاضر این کار را در دستگاه‌های دارای دوربین دوگانه خود انجام می‌دهد. سیستم مشابه دیگر اپل، فیس آیدی است که با ایجاد یک نقشه دقیق از صورت کاربر، آن را از میان میلیون‌ها نفر تشخیص می‌دهد.

سیستم مشابه دیگر، در دوربین‌های پیکسل‌های جدید گوگل دیده می‌شود؛ اما سنسور سه‌بعدی مبتنی بر لیزر آتی اپل به مراتب قوی‌تر و دقیق‌تر خواهد بود. پیش از معرفی فیس‌آیدی نیز چندین گوشی‌ساز همچون سامسونگ از سیستم تشخیص چهره مخصوص خود در گوشی‌های موبایلشان استفاده می‌کردند، اما ویژگی تکنولوژی تشخیص چهره فیس‌آیدی، پیشرفته‌تر بودن آن است.

در برترین آیفون سال ۲۰۱۹، فیس‌آیدی همچنان حضور خواهد داشت؛ اما در پشت دستگاه نیز یک سنسور سه‌بعدی مبتنی بر لیزر به کار می‌رود که کاربرد مخصوص به خود را دارد.

اپل علاقه و توانایی خود را در تکنولوژی واقعیت افزوده به‌خوبی نشان داده است. یکی از اصلی‌ترین و جدیدترین ویژگی‌های آی‌اواس ۱۱ به‌عنوان نسخه سال ۲۰۱۷ پلتفرم موبایل اپل، قابلیت‌های مربوط به واقعیت افزوده است. این سیستم‌عامل در حاضر توانایی بسیار بالایی در این تکنولوژی دارد، اما با افزوده شدن یک سنسور سه‌بعدی مبتنی بر لیزر و توسعه دستگاه به‌صورت سخت‌افزاری، شاهد جهشی بزرگ در دنیای واقعیت افزوده خواهیم بود.

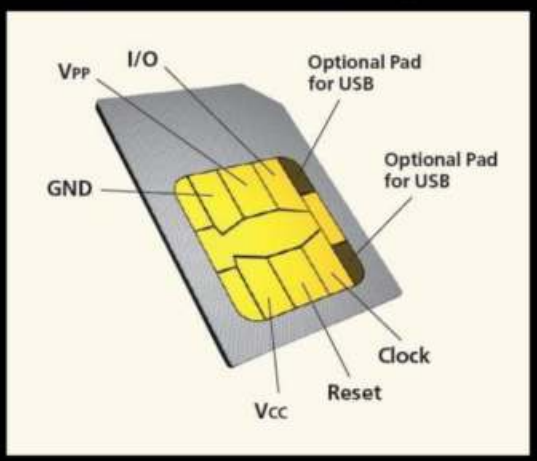


سیم کارت، یک کارت هوشمند، با دنیای پیچیده‌ی درونش، خیلی شناخته شده نیست. شاید بسیاری از شماها تا به‌حال فکر نکرده بودید که درون سیم کارت چند سانتی متری، چه دنیایی وجود دارد. در این مقاله قصد داریم که یک گوشه از دنیای درون سیم کارت را به شما نشان دهیم.

پس با ما همراه باشید...

- **تعریف مقدماتی سیم کارت**

ابتدا به سراغ قسمت سخت افزاری می‌رویم. سیم کارت ها ابزارهایی سخت افزاری هستند که تلفن‌های ما را قادر به برقراری ارتباط با اپراتورها می‌کنند و فضای بسیار کمی نیز برای ذخیره شماره‌ها و اطلاعات خود دارند. حال اگر بخواهیم بصورت تخصصی آن‌ها را بررسی نماییم، ابتدا نگاهی می‌کنیم به ظاهر سخت افزاری سیم کارت و تشریح بخش‌های مختلف آن:



VCC: منبع تغذیه

RST: ری‌استارت ارتباط و سیگنال

CLK: ایجاد سیگنال کلاک برای سیم کارت

GND: اتصال زمین

VPP: اختصاص یافته برای ولتاژهای برنامه نویسی

I\O: ورودی و خروجی

C4 -C8: که به رنگ تیره مشخص شده اند و برای ارتباطات خارجی از طریق یو اس بی و دیگر کاربردها طراحی شده اند.

یک سیم کارت شامل یک بخش چیپ می‌باشد که در واقع یک مموری ROM از ۶۴ تا ۵۱۲ کیلوبایت می‌باشد که سیستم عامل بر روی این بخش قراردارد و در بخش EEPROM اطلاعات کاربری مانند تماس‌ها و اس ام اس و.... ذخیره می‌شود. بسیاری از این سیم کارت‌ها با ولتاژ ۱,۸ – ۳ تا ۵۵ ولت کار می‌کنند. کارخانه‌های تولید کننده سیم کارت غالبا از سه الگوریتم برای تولید سیم کارت استفاده می‌کنند:

۱) COMP128v1
۲) COMP128v2
۳) COMP128v3

امروزه نیز صرفا سیم کارت‌های COMP128v1 قابل کلونینگ یا کپی می باشند، چرا که فعلا صرفا الگوریتم این نوع از سیم کارت کشف شده و هنوز در مورد دیگر الگوریتم‌ها اطلاعات چندانی در دسترس نمی‌باشد و قابل ذکر است که ۷۰٪ سیم کارت‌های امروزی از این الگوریتم بهره می‌برند.

- **بخش نرم افزاری سیم کارت**

ساختار زیر یک چارت کلی از فایل سیستم‌های موجود داخل سیم کارت است و این فایل سیستم‌ها ،که تعدادی از آن‌ها اجباری و تعدادی اختیاری هستند ، هر کدامشان یک وظیفه را باید انجام بدهند که این وظایف شامل: ذخیره شماره سریال منحصر به فرد برای هر کدام از سیم کارت‌ها، ذخیره اطلاعات در مورد سرویس‌هایی که یک سیم کارت می‌تواند آن‌هارا به اجرا در بیاورد. البته که این قابلیت سیم کارت ها هم وابسته به نوعشان است که می‌تواند، sim یا usim باشد که اولی برای نسل دوم سیم کارت‌هاست و دومی برای نسل سوم سیم کارت‌ها(البته تفاوت‌های دیگری هم دارد)، مورد بعدی در نوع سیم

کارت‌ها است اگرآن را شنیده باشید usim R5 و ... عددهای دیگه‌ای که پشت سر ای R که آن ها بسته به سال تولید و توانایی هایشان متفاوت هستند و برای یافتن اطلاعات

حال فرض کنید یک سیم کارت ازنوع usim R5 داریم (که معمولا اکثر سیم کارت های موجود در بازار ازاین نوع است)ومیخواهیم درون آن را بررسی کنیم:

شکل بالا ساختار کلی همان فایل سیستم ها را نشان میدهد (قبل از اینکه دنیای درونش را بررسی کنیم ،شاید بهتر باشد بدانید که چه طور درون هر کدام از این هارا می شود خواند و یا اینکه اصلا می شود در این فایل هاتغییرات ایجاد کرد ؟ در جواب سوالتان باید بگوییم که شرکتی به نام GEMALTO با استفاده از نرم افزاری ویژه این اطلاعات را بدست می آورند)

طبق چارتی که مشاهده می کنید، در اولین بخش MF نوشته

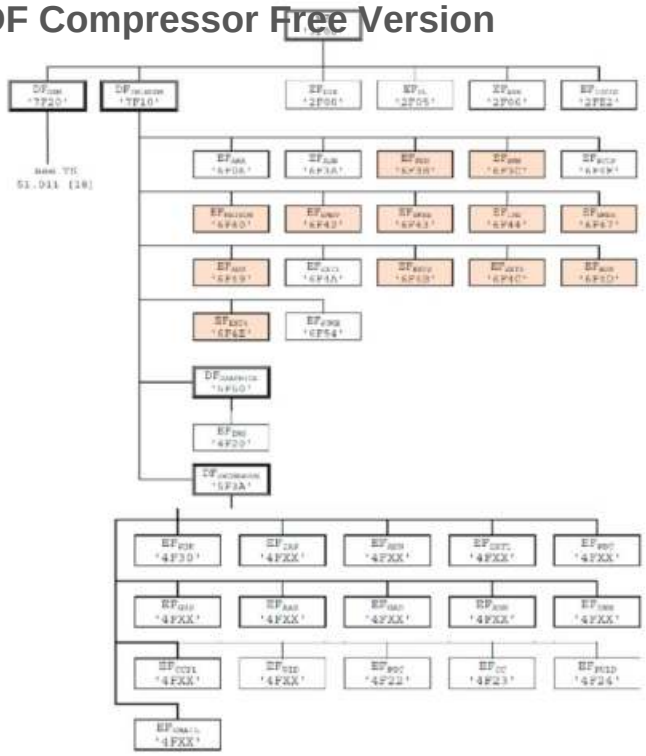
Table 13.1: EF_{DIR} at MF-level

Identifier: '2F00'	Structure: Linear fixed		Mandatory
SFI: Mandatory			
Record size: X bytes		Update activity: low	
Access Conditions:			
READ	ALW		
UPDATE	ADM		
DEACTIVATE	ADM		
ACTIVATE	ADM		
Bytes	**Description**	**M/O**	**Length**
1 to X	Application template TLV object	M	X bytes

شده است یعنی Master file که اولین فایل هست و تمام فایل های دیگر را درون خود دارد واما این فایل شامل سه تا DF و چها رتا EF است که DF(Dedicated File) ها هم هرکدام شامل یکسری EF (Elementary Files) است.

در هر کدام از این EF ها که یک فایل وجوددارد(در داخل همان قطعات سخت افزاری که بالا ذکرشد، ذخیره شده اند) اطلاعاتی را ذخیره کرده است.برای مثال ؛اولین فایل سیستمی که می بینید ، که مستقیما هم متصل به MF هست ، مربوط به اپلت هایی است که داخل این سیم کارت فعال شده اند و زمانی که شما یک اپلت را بخواهید با نرم افزار بسازید، به شما یک AID می دهد و این فایل این اطلاعات را نگه می دارد و اگر اپلتی بخواهد اجرا شود ابتدا سراغ این فایل می رود و اگر AID شناخته شده باشد اجرا می شود .

فرصت نیست که حدود ۲۰۰ فایل سیستم که یکسری از آن ها اجباری و یکسری غیر اجباری هستند را یک به یک توضیح دهیم، اما ؛ موارد دیگری که خوب است به آن ها اشاره شود شامل: فایل سیستمی که برای ارسال هر اس ام اس فعال می شود و با کمک آن میتوان اس



ام اس را فرستاد ، فایل سیستم هایی که مربوط به دفترچه تلفن هستند ،فایل سیستم مربوط به اطلاعات منحصر به فرد سیم کارت ،که برای هر سیم کارت تنها یک شماره و آن شماره تنها برای همان سیم کارت است و...

برای نمونه جدول زیر ، اطلاعات دسترسی به اطلاعات درون یک فایل سیستم مشخص است را نشان می دهد .

در مورد سطوح دسترسی که در مقابل هر کدام از کنش های لازم ، نوشته شده است داریم :

ADM: تخصیص این سطوح مسئولیت افرادی است که این کارت را صادر کرده اند .(ارائه دهنده کارت یا ارائه کننده تلفن که کارت را به مشترکین خود می دهد.)

NEV: دستور هرگز در فایل اجرا نمی شود؛

ALW: دستور همیشه در فایل اجرا می شود؛

تأیید اعتبار کارت ۱ - (CHV1) این وضعیت پس از تأیید صحت PIN1شخصی کاربر امکان دسترسی به فایل ها را فراهم می کند یا اگر تأیید PIN1 غیرفعال باشد.

تأیید اعتبار کارت ۲ - (CHV2) این وضعیت پس از تایید موفقیت آمیز PIN2 کاربر امکان دسترسی به فایل ها را فراهم می کند و یا اگر تأیید PIN2 غیر فعال باشد.

و در مورد هر کدام از کد های ذکر شده در بالا داریم:

pin:(Personal Identification Number)شماره شناسایی شخصی- ۲ پین وجود دارد (پین ۱ و PIN2) یک PIN (شناسه شخصی) یک کد عبور ۴-۸ رقمی است که برای تأیید هویت کاربر به یک سیستم استفاده می شود.

پین تلفن یک دستگاه ذخیره سازی داخل سیم کارت است و همچنین یک اقدام امنیتی برای محافظت از سیم کارت به سرقت رفته است.اگر "عملکرد امنیتی PIN" توسط گوشی شما پشتیبانی می شود و فعال شده است، هر بار که شما تلفن خود را شروع می کنید، باید PIN را برای باز کردن قفل سیم کارت وارد کنید.هنگامی که هر دو پین و فعال هستند، ابتدا باید PIN و سپس رمز عبور قفل صفحه را برای باز کردن قفل گوشی خود را وارد کنید.

پین توسط اپراتور مخابراتی ارائه می شود و می تواند مجددا تنظیم و اصلاح شود.

نکته : اگر پین بیش از سه بار اشتباه وارد شود، هر دو سیم کارت و تلفن قفل خواهد شد. آنها می توانند با وارد کردن PUK باز شوند.

اگر پین را نمی دانید و یا آن را فراموش کرده اید، برای دستیابی به اپراتور مخابراتی با ما تماس بگیرید (به طور کلی، ۱۲۳۴ یا ۰۰۰۰ پین پیش فرض برای سیم کارت های سیم کارت است، اما فقط برای مرجع، با دقت استفاده کنید).

Puk: یک PUK (کلید باز شناسایی شماره شخصی) نیز PUC (Pin Unlock Code) نامیده می شود.

puk یک رشته از هشت اعداد نامنظم است که نمی توانند توسط کاربر تنظیم مجدد یا اصلاح شوند. Pukفقط زمانی تغییر می کند که سیم کارت جایگزین شود.

در نظر داشته باشید که puk برای باز کردن قفل پین استفاده می شود و برخی از PUK ها همراه با سیم کارت هایی هستند که کاربر خریداری می کند.

اگر PUK ، ده بار در یک ردیف اشتباه وارد شود ،هر دو سیم کارت و تلفن به طور دائم قفل شده و شما باید گواهینامه معتبر خود را به فروشگاه اپراتورهای مخابراتی ببرید تا یک سیم کارت جدید دریافت کنید . بنابراین، هنگامیکه سیم کارت و تلفن با پین قفل می شوند، شما باید به طور همزمان در تماس تلفنی با اپراتور مخابراتی باشید.

پس از تایید ساده برای داده های کاربر، می توانید PIN یا PUK را برای باز کردن قفل سیم کارت و تلفن خود بدست آورید.

ADMهم یک سطح دسترسی است که همان طور که گفته شد فقط اپراتور و سازنده ، آن را دارد و این یک کد است که اپراتور با زدن این کد می تواند از این سطح دسترسی استفاده نماید .

همچنین اطلاعاتی دیگری که این فایل ها در اختیار ما می گذارند شامل:

International Mobile Subscriber Identity or IMSI: عددی ۱۳ تا ۱۵ رقمی می باشد که برای شناسایی اپراتور استفاده میگردد. ۳ رقم اول کشور اپراتور و ۳ رقم بعدی خود اپراتور را نشان میدهد و اعداد بعدی نشان دهنده اطلاعات و جزییات اپراتور میباشد این عدد بر روی لاشه سیم کارت قابل مشاهده است.

Authentication Key or Ki : کیلدی بین اپراتور و کاربر سیم کارت میباشد که به اپراتور اجازه میدهد تا کاربر را شناسایی و اطلاعات و هویت وی را احراز نماید .این اطلاعات در حین صدور سیم کارت از کاربر گرفته شده و به این کلید اختصاص می یابد و باید با روش های خاص و از طریق خواننده کارت بازیابی شود.

Location Area Identity or LAI:اطلاعاتی درباره موقعیت مکانی ،که هر سیم کارت ،اخرین اطلاعات موقعیت مکانی بر حسب BTSوصل شده را در خود ذخیره میکند . در دیگر بخش ها نیز پیام های اس ام اس و شماره تلفن ها و تماس ها ذخیره میشوند.

خوب شاید در ذهنتان هزاران سوال ایجاد شده باشد، چرا که این موضوع خیلی آشنا نیست، اما برای آشنایی بیشتر توصیه می کنم در استاندارد های ETSI به دنبال پاسخ سوالاتتان بگردید .

از این سیم کارت ها و اطلاعات داخل آن ، می توان استفاده های زیادی کرد ؛از کیف پول های دیجیتال که سیم کارت بتواند اطلاعات را نگه دارد ،تا اپلت های over the air که بدون دسترسی های مستقیم اپراتور، کار هایی روی سیم کارت به صورت over the air انجام می دهد.

تعمیرات مهندس

Loadنشدن search در ویندوز 10

حدودا یک ماه بود که search ویندوز بالا نمی‌آمد و من برای پیدا کردن برنامه‌هایم به مشکل برخورده بودم. خیلی هم برای رفع مشکل تلاش کردم و زمان زیادی صرف رفع مشکل کردم اما درست نشد؛ تا اینکه یک فکری به ذهنم رسید.!

خوب؛ حتما هزاران نفر هم این مشکل را داشته‌اند و از آن هزار نفر، حداقل ۵ نفر ، مشکلش را حل کرده‌اند و جواب را با بقیه به اشتراک گذاشته اند. پس، در اینترنت گشتم و راه حل را پیدا کردم .

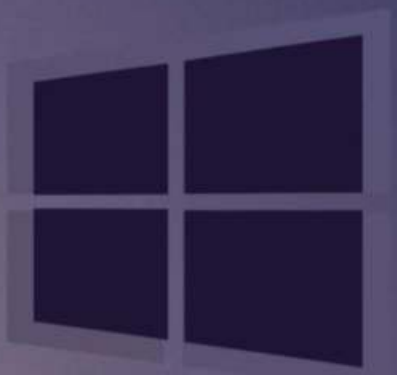
- **search ویندوز 10با مشکل همراه است؟**

Search ویندوز 10 همراه با یک دستیار شخصی به نام "کورتانا " ارائه می شود. عامل اصلی اختلال مربوط به تنظیمات کورتانا و لینک شدنش به حساب مایکروسافت است. اما راه حل این مشکل چیست ؟؟

- **راه حل رفع مشکل loadشدن search در ویندوز10چیست؟**

راه حل اول: کورتانا را غیر فعال کنید! برای این کار بر روی نوار وظیفه (taskbar) کلیک راست کنید و گزینه TaskManager را انتخاب کنید. مطمئن شوید که در بخش Processes قرار دارید. سپس دنبال Cortana یا search بگردید.روی آن راست کلیک کرده وسپس آن را غیر فعال کنید. مشکل حل می‌شود اما شما نمی‌توانید از دستیار شخصی کورتانا استفاده کنید.

راه حل دوم: برای این که Search در ویندوز 10 درست عمل کند، باید مطمئن شوید که Windows Search service فعال شده باشد. برای فعالسازی، روی کیبورد خود همزمان دکمه Windows و حرف R را فشار دهید تا قابلیت Run باز شود. بعد از آن، عبارت services.msc را تایپ کنید تا صفحه Services باز شود. از این جا، Windows Search service را پیدا کنید. سپس، روی آن راست کلیک کنیدو عبارت properties را انتخاب کنید. وقتی صفحه Windows Search Properties باز شد، عبارتStartup Type کلیک کنید و آن را در حالت Automatic قرار دهید. پس از آن، روی OK کلیک کنید تا تغییرات ذخیره شود.



راه حل سوم: چنان چه با انجام دو مرحله بالا Search ویندوز شما مجددا load نشد، با روش ذیل باید درست شود. ابتدا Control Panel را بیاورید. عبارت Indexing Options را جستجو کنید وآن را کلیک کنید تا صفحه‌ای در برابر شما باز شود. روی Troubleshoot search and indexing در زیر همان صفحه کلیک کنید و مراحلش را دنبال کنید.اگر بلافاصله مشکلتان درست نشد کامپیوتر را resetکنید.

آیا تعمیری این دفعه برایتان مفید بود؟

تعمیری‌های خود را برای ما به ایمیل مجله پردازش ارسال کنید .تعمیری شماره بعد مجله : آیا تاچ پد لپتاپ شما کار نمی کند؟