

## سوال اول

**الف)** سیستم عامل یا (OS) Operating System بخش نرم افزاری اصلی کامپیوتر شماسست که وظیفه‌ی مدیریت اپلیکیشن‌ها و برقراری ارتباط آن‌ها با سخت افزار را بر عهده دارد؛ به این صورت که اپلیکیشن درخواست‌های خود را به سیستم عامل می‌دهد و سیستم عامل آن‌ها را مدیریت می‌کند. تصور کنید هر نرم افزاری که توسعه داده می‌شود نیاز داشت که به صورت مستقیم با سخت افزار کامپیوتر در ارتباط باشد، در این صورت نیاز بود برای انواع سخت افزارها برنامه‌های جداگانه‌ای توسعه داده شوند. اما سیستم عامل با استفاده از درایورهای مربوط به سخت افزار وظیفه‌ی برقراری ارتباط نرم افزار و سخت افزار را بر عهده می‌گیرد پس در این صورت می‌توان نرم افزارها را بر اساس انواع سیستم عامل‌ها توسعه داد. سیستم عامل وظایفی نظیر مدیریت فایل‌ها، مموری، پردازشگر، ورودی و خروجی‌ها و کنترل کردن سخت افزارهای جانبی که به کامپیوتر متصل می‌شوند را دارد.

سیستم عامل‌ها تنها مختص به کامپیوترهای شخصی یا لپ تاپ شما نیستند و امروزه در بسیاری از لوازم الکترونیکی نظیر موبایل، تلویزیون، کنسول‌های بازی و ساعت‌های هوشمند انواع مختلف آن‌ها استفاده می‌شود.

از سیستم عامل‌های شناخته شده می‌توان به Windows، macOS، Linux، Android و iOS اشاره کرد.

**ب)** کامپیوترهای اولیه در واقع mainframe هایی بودند که از هیچ سیستم عاملی بهره نمی‌بردند. هر کامپیوتر تنها می‌توانست در هر لحظه یک کار را برای یک کاربر در مدت زمان محدودی انجام دهد انجام دهد. برنامه و داده مورد نیاز توسط یک سری کاغذهای سوراخ دار و یا نوارهای مغناطیسی یا کاغذی به کامپیوتر داده می‌شد تا برنامه اجرا شود و سپس به پایان برسد یا متوقف شود. این برنامه‌ها توسط پنل هایی که حاوی مقدار زیادی سویچ فیزیکی و چراغ بودند اشکال زدایی می‌شدند.

اولین سیستم عامل در سال ۱۹۵۶ توسط جنرال موتورز برای کامپیوتر IBM ارائه شد. کار اصلی آن این بود که یک برنامه را به صورت خودکار پس از اتمام یک برنامه‌ی دیگر آغاز کند. بسیاری معتقدند کامپیوتر اطلس دانشگاه منچستر انگلستان در سال ۱۹۶۲ را می‌توان نخستین استفاده از یک سیستم عامل مدرن دانست که پیشرفت قابل توجهی برای به وجود آمدن سیستم عامل‌های امروزی بود.

## سوال دوم

CPU : مخفف شده‌ی Central Processing Unit به معنای واحد پردازش مرکزی که به مغز رایانه معروف است.

این واحد بخشی از رایانه است که وظیفه‌ی اجرای دستورالعمل‌ها را بر عهده دارد. اجرای این دستورالعمل‌ها می‌تواند شامل پردازش و انجام عملیاتی مانند عملیات محاسباتی (arithmetic)، منطقی (Logical)، کنترلی (controlling) و ورودی و خروجی (Input/Output) باشند.

GPU : مخفف شده‌ی Graphical Processing Unit به معنای واحد پردازش گرافیکی است و مسئولیت نمایش تصاویر و ویدئوها را بر عهده دارد.

پردازنده گرافیکی روی کارت‌های گرافیکی قرار می‌گیرد و به حافظه RAM نیاز دارد؛ روی کارت‌های حافظه اختصاصی برای کارهای گرافیکی وجود دارد که به اختصار به آن (GDDR) می‌گویند علاوه بر این کارت گرافیکی، ابزارهای دیگری نیز مانند پورت‌ها برای انتقال اطلاعات به بقیه اجزای سیستم و کولر نیز دارند.

GPU علاوه بر پردازش گرافیک و تصویر در یادگیری ماشین، رمزنگاری و سریع‌تر کردن محاسبات به CPU کمک میکند و در مواقعی که CPU توان اجرای تمامی عملیات‌ها را ندارد GPU بخشی از عملیات‌ها را انجام می‌دهد.

GPU به دلیل تعداد هسته‌های زیادی که دارد می‌تواند تعداد زیادی عملیات را به صورت موازی پیش ببرد اما CPU قادر به انجام دادن تعداد محدودی عملیات است زیرا تعداد بسیار کم‌تری هسته نسبت به پردازنده گرافیکی دارد ولی در عوض قوی‌تر هستند.

## سوال سوم

RAM: حافظه ی اصلی کامپیوتر محسوب می شود و یک حافظه ی فرار است. یعنی، اطلاعات به طور موقت در آن ذخیره شده و با قطع جریان برق از روی RAM پاک می شوند. این حافظه به علت بسیار سریع بوده ولی به علت محدودیت ذخیره سازی موقتی آن، نمی تواند به تنهایی پاسخگوی نیازهای یک سیستم باشد. همچنین حافظه ی RAM ظرفیت کمی را در حجم محدود فراهم می کند.

HDD یا دیسک سخت: دیسک سخت حافظه ی غیر فرار محسوب شده و با قطع جریان برق، اطلاعات را از دست نمی دهد. دیسک سخت با استفاده از چرخش دیسک و حرکت فیزیکی یک بازو روی آن به نوشتن و خواندن و دسترسی اطلاعات می پردازد که گلوگاه سرعت آن می باشد و مسئول سرعت پایینتر این حافظه است. همچنین به علت حرکت فیزیکی بازو گرما و صدای بیشتری تولید می کند. HDD از SSD بسیار ارزان تر است و به دلیل قیمت مناسب آن هنوز در کامپیوترها با ظرفیت بالا استفاده می شود.

SSD یا حالت جامد: حافظه های SSD به علت استفاده از تکنولوژی جدیدتر، قیمت بالایی دارند و در نتیجه معمولاً با ظرفیت های کمتری در دسترس هستند. عدم نیاز به حرکت فیزیکی در آن باعث کاهش تولید صدا و گرما هنگام استفاده و افزایش طول عمر آن در دراز مدت است.

هزینه	سرعت	ظرفیت	نحوه ذخیره داده	نوع حافظه	
زیاد 10\$ pgb	زیاد	کم	ذخیره سازی با استفاده از سیستمی از ترانزیستورها و خازن ها	موقت	RAM
کم 0.04-0.06\$ pgb	کم	زیاد	ذخیره سازی مغناطیسی روی دیسک های چرخان	دائم	HDD
متوسط 0.1\$ pgb	متوسط	متوسط	ذخیره سازی روی حافظه ی فلش	دائم	SSD

## سوال چهارم

**الف)** همان‌طور که میدانید خیلی از مفاهیمی که ما به زبان ساده بیان می‌کنیم برای کامپیوتر قابل‌درک نیستند و لازم است به زبان ماشین تبدیل شوند. یکی از این مفاهیم کاراکترهای مختلفی هستند که از آن‌ها استفاده می‌کنیم. با استفاده از **character encoding** به هر کاراکتر تعدادی صفر و یک نسبت می‌دهیم تا آن کاراکتر برای ماشین قابل‌فهم باشد. برای نسبت دادن این صفر و یک‌ها به هر کاراکتر استانداردهای مختلفی در طول زمان معرفی شدند. از استانداردهای شناخته‌شده می‌توان **ASCII** و **Unicode** را نام برد. در **ASCII** به هر کاراکتر ۷ بیت اختصاص می‌یابد که با استفاده از آن می‌توان  $2^7 = 128$  کاراکتر را نمایش داد که برای تمام حروف بزرگ و کوچک انگلیسی و اعداد و برخی کاراکترهای دیگر کافی بود. اما از آنجاکه کامپیوترها هر ۸ بیت (یک بایت) را پردازش می‌کردند لازم بود از بیت هشتم هم در **ASCII** استفاده شود که در این صورت می‌توان ۲۵۶ کاراکتر را نمایش داد. برای کاراکترهای بعد از ۱۲۸ هیچ‌گاه استاندارد مشخصی تعریف نشد و در هر منطقه‌ای بر اساس نیاز خودشان شخصی‌سازی می‌شد. درنهایت ۱۵ استاندارد مختلف معرفی شد که هرکدام به نحوی کاراکترهای بعد از ۱۲۸ را نمایش می‌دادند.

پس از مدتی **Unicode** معرفی شد که دیگر محدودیت‌های **ASCII** را نداشت و قادر بود با اختصاص دادن تعداد بیت‌های بیشتر تقریباً هر کاراکتری در دنیا را نمایش دهد (بیش از ۱۴۰۰۰۰ کاراکتر). مشکلی که در ابتدا با آن مواجه بودند ارسال و دریافت این کاراکترها بود زیرا گاهی لازم بود برای یک کاراکتر انگلیسی که تنها به ۷ بیت نیاز داشت ۳۲ بیت ارسال و دریافت شود که بهینه نبود. بنابراین استانداردهای **UTF-16**، **UTF-32** و **UTF-8** معرفی شدند که عدد آخر هر کدام نشان‌دهنده‌ی تعداد بیت‌های استفاده‌شده در آن است. در حال حاضر **UTF-8** پرکاربردترین استاندارد است که اکثر مشکلاتی که تاکنون با آن مواجه بودند نظیر اشغال کردن فضای مناسب برای هر کاراکتر و نمایش دادن همه کاراکترها را برطرف می‌کند.

**ب)** با توجه کردن به محدودیت ارسال پیام کوتاه در تلفن‌های خود به این موضوع پی می‌بریم که محدودیت کاراکترهای انگلیسی برای ارسال یک پیام ۱۶۰ کاراکتر و محدودیت ارسال پیام‌های فارسی ۷۰ کاراکتر است که این نسبت به **character encoding** مربوط به هر کدام برمی‌گردد. در کل برای کاراکترهای پرکاربرد (کاراکترهای انگلیسی و برخی علامت‌های پرکاربرد) در تلفن‌های همراه از استاندارد **GSM-7** استفاده می‌شود که برای هر کاراکتر ۷ بیت در نظر می‌گیرد و تا ۱۶۰ کاراکتر با استفاده از آن می‌توان ارسال کرد. برای کاراکترهای خارج از این محدوده که معمولاً کاراکترهای زبان‌های غیر انگلیسی را شامل می‌شوند از استاندارد **USC-2** استفاده می‌شود که ۱۶ بیت برای هر کاراکتر اختصاص می‌دهد. این نسبت ۷ به ۱۶ با تعداد کاراکترهای مجاز در هر پیام همخوانی دارد.

**پ)** همان‌طور که در قسمت‌های قبل توضیح داده شد با استفاده از استاندارد **UTF** می‌توان گستره‌ی بسیار بزرگ‌تری از کاراکترها را نمایش داد درحالی‌که با **ASCII** تنه‌ای قابلیت نمایش ۱۲۸ کاراکتر به صورت ثابت وجود داشت. تشابه این دو استاندارد در این است که در هر دو آن‌ها ۱۲۸ عدد اول به کاراکترهای یکسانی اشاره می‌کنند.

## سوال پنجم

در معماری ۳۲ بیت خانه‌های حافظه (رجیسترها) در CPU دارای ۳۲ بیت و در نتیجه ظرفیت ذخیره سازی ۲ به توان ۳۲ مقدار متفاوت را دارند که در نتیجه می‌توانند تنها از ۴ گیگابایت رم استفاده کنند در حالی که معماری ۶۴ بیت میلیون‌ها برابر بیشتر بوده و امکان استفاده از بیش از ۴ گیگابایت رم را فراهم می‌سازد.

الف) به صورت کلی در سیستم‌هایی که معماری ۶۴ بیت دارند می‌توان از برنامه‌های ۳۲ بیتی استفاده کرد اما به دلیل عدم استفاده از حداکثر منابع، سرعت آن‌ها به مراتب از ورژن ۶۴ بیت کمتر می‌باشد.

ب) با توجه به توضیحات بیان شده این امر امکان پذیر نمی‌باشد.

پ) ۶۴ بیت:

### Linux

```
saman@Snapp:~  
saman@Snapp ~$ uname -m  
x86_64  
saman@Snapp ~$
```

### Windows

#### About

Your PC is monitored and protected.

[See details in Windows Security](#)

#### Device specifications

Device name	DESKTOP-AVJ308S
Processor	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.59 GHz
Installed RAM	16.0 GB (15.9 GB usable)
Device ID	7ED42D0A-C390-4083-B366-CD4A391FF338
Product ID	00328-90000-00000-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

## سوال ششم

**الف) Compiler** یا کامپایلر: کامپایلر وظیفه تبدیل یک زبان برنامه نویسی معمولاً سطح بالا به زبان ماشین را انجام می‌دهد. کامپایلر تمام برنامه را یکجا دریافت می‌کند و آن را به زبان ماشین ترجمه می‌کند. کامپایلرها معمولاً برای اینکار ابتدا کد را به یک کد میانی تبدیل می‌کنند که باعث می‌شود حافظه‌ی بیشتری از مفسرها اشغال کنند. همچنین کامپایلرها حین این پروسه کد برنامه را بررسی می‌کنند و بخشی از غلط‌های آن که مربوط به پیش از اجرای برنامه می‌باشد را نشان می‌دهد و پس از آن در صورت نیاز با کدهای دیگر لینک می‌شود. پس از یک کامپایل موفقیت آمیز، برای اجرای متعدد برنامه نیازی به کامپایل مجدد آن نیست.

مانند زبان های برنامه نویسی C و C++

**Interpreter** یا مفسر: یک مفسر برخلاف کامپایلر عمل تبدیل به کد ماشین را حین اجرای برنامه و به صورت ترجمه و اجرای یک خط در هر زمان انجام می‌دهد. بنابراین برخلاف کامپایلر که غلط‌هایی را در زمان کامپایل برنامه متوجه می‌شود، مفسر تمام غلط‌ها را در حین اجرای برنامه اعلام می‌کند. بنابراین برنامه‌نویسان باید پس از رفع ارورهای قبل تازه با ارورهای جدید رو به رو می‌شوند که بهینه کردن کد را برای برنامه‌نویسان سخت می‌کند.

مانند زبان‌های برنامه‌نویسی Ruby و Python

**ب) High-level Language** یا زبان سطح بالا، به زبان برنامه نویسی ای اطلاق می‌شود که از زبان ماشین فاصله گرفته و به زبان انسان نزدیکتر است و فهم و استفاده از آن برای ما به عنوان برنامه‌نویس آسان‌تر است. زبان‌های سطح بالا معمولاً با استفاده از زبان‌های سطح پایین‌تر نوشته و ساخته شده‌اند و به مرور این لایه‌های مختلف تجرید که کمک بزرگی به سادگی و کاهش پیچیدگی برنامه نویسی می‌کنند، برنامه نویس را برای رسیدن به حداکثر کارایی محدود می‌کنند. بنابراین در کمال تعجب زبان‌های سطح بالا نسبتاً کارایی پایین‌تری از لحاظ سرعت و استفاده از حافظه دارند.

با این حال بسیاری از برنامه‌های کنونی به سطحی از کارایی که در بالا راجع به آن صحبت کردیم حساس نیستند و در نتیجه امروزه به صورت گسترده‌ای از زبان‌های سطح بالا در موارد مختلف استفاده می‌شود.

نمونه زبان‌های سطح بالا: C, java, python

**Low-level Language** یا زبان سطح پایین، به زبان برنامه‌نویسی ای گفته می‌شود که به زبان ماشین نزدیک‌تر است و این یادگیری و استفاده و debug کردن آن را دشوار می‌کند.

هنگام کار با زبان‌های سطح پایین به علت نزدیکی آن به زبان ماشین و رابطه‌ی نزدیک آن با سخت‌افزار، باعث می‌شود برنامه‌نویس درک بهتری از سخت‌افزار داشته باشد و برنامه‌نویس حرفه‌ای با کنترل بیشتری به استفاده بهینه از حافظه و افزایش کارایی و سرعت برنامه بپردازد.

زبان‌های سطح پایین در کاربردهای محدودی استفاده می‌شوند که به این میزان کنترل روی برنامه‌نویسی نیاز دارد.

نمونه زبان‌های سطح بالا: Assembly

## سوال هفتم (امتیازی)

Linker یا Link editor یک برنامه سیستمی است که object file های ایجاد شده توسط compiler و assembler را دریافت کرده، یک فایل قابل اجرا، کتابخانه و یا یک object file واحد تحویل می دهد (این عمل با جمع آوری و اضافه کردن کتابخانه های استفاده شده در کد همراه است).

انواع Linker:

- **Static:** در این نوع، linker در زمان کامپایل شدن کد تمامی کتابخانه های استفاده شده را مستقیماً کپی کرده و در فایل خروجی قرار می دهد، که سبب می شود حجم کلی خروجی افزایش یابد اما باعث می شود احتمال بروز مشکل کاهش یابد. (چرا که نیازی به وجود کتابخانه های مورد استفاده در سیستم نمی باشد) همچنین سرعت اجرای فایل خروجی افزایش می یابد چرا که در زمان اجرا تمامی موارد مورد نیاز موجود می باشند و نیازی به Load کردن فایل ها و کتابخانه ها نمی باشد.
- **Dynamic:** این Linker ها بجای کپی کردن کتابخانه ها و آوردن مستقیم آن ها در کد و فایل خروجی، تنها نام آن ها را قرار داده و در زمان اجرا آن ها را بارگذاری می کند. این امر سبب کند شدن برنامه می شود ولی از مصرف حافظه اضافی جلوگیری می کند. همچنین نیاز است که کتابخانه های مورد نیاز روی سیستم موجود باشند تا بتوان از خروجی آن استفاده کرد.