Configuration Script

اسکریپت پیکربندی ابزار قدرتمندی است که برای خودکارسازی راه اندازی و سفارشی سازی سیستم

های نرم افزاری استفاده می شود. این اساسا یک اسکریپت است که اغلب به زبان برنامه نویسی مانند Bash، Python یا PowerShell نوشته می شود و مجموعه ای از دستورات را برای پیکربندی

یک سیستم بر اساس پارامترهای خاص اجرا می کند

چرا از اسکریپت های پیکربندی استفاده کنیم؟

1 اتوماسیون

تلاش دستی و خطای احتمالی انسانی را کاهش می دهد

پیکربندی های منسجم را در چندین سیستم تضمین می کند.

2 انعطاف پذیری

امکان سفارشی سازی و سازگاری آسان با محیط های مختلف را فراهم می کند

می تواند برای کنترل سناریوهای مختلف پارامتری شود

3 تکرارپذیری

فرآیند راه اندازی سیستم های جدید یا پیکربندی مجدد سیستم های موجود را ساده می کند

4 کنترل نسخه

اسکریپت های پیکربندی را می توان بر اساس نسخه کنترل کرد و ردیابی تغییرات و بازگشت به

تنظیمات قبلی را آسان تر می کند

موارد استفاده رایج

نصب نرم افزار

دانلود و نصب بسته های نرم افزاری

پیکربندی تنظیمات و وابستگی های نرم افزار

پیکربندی سیستم

راه اندازی رابط های شبکه، قوانین فایروال و سرویس های سیستم

سفارشی کردن تنظیمات در سراسر سیستم مانند حساب های کاربری و مجوزها

استقرار برنامه

استقرار و پیکربندی برنامه های کاربردی وب، پایگاه های داده و سایر نرم افزارها

تامین سرور

ایجاد خودکار سرورهای جدید، از جمله نصب سیستم عامل و پیکربندی اولیه.

Test Automation

در دنیای امروز، توسعه نرم‌افزار به سرعت در حال پیشرفت است و نیاز به تضمین کیفیت نرم‌افزار بیش از پیش احساس می‌شود. تست خودکار (Test Automation) یکی از روش های ماشین برای بهبود کیفیت و سرعت توسعه نرم افزار است. در این مقاله به بررسی مفاهیم، ​​مزایا، چالش‌ها و ابزارهای تست خودکار پرداخت

مفهوم تست خودکار

تست فرآیندهای خودکار است که در آن نرم افزارهای خودکار به صورت خودکار اجرا می شوند. این روش به جای تکیه برای تست های دستی، از ابزارها و اسکریپت های نرم افزاری برای اجرای تست ها استفاده می کند. این امر باعث افزایش سرعت، دقت و پوشش تست می شود

سیستم تست خودکار

**1افزایش سرعت:** تست‌های خودکار بسیار سریع‌تر از تست‌های دستی اجرا می‌شوند. این امر به تیم‌های توسعه اجازه می‌دهد تا سریع‌تر به بازخورد تست‌ها دسترسی پیدا کنند و مشکلات را زودتر شناسایی کنند.

**2افزایش دقت:** تست‌های خودکار کمتر مستعد خطاهای انسانی هستند. این امر باعث افزایش و کاهش تعداد باگهایافزاری می شود

**3افزایش پوشش:** تست‌های خودکار می‌توانند به طور مستقیم از نرم‌افزار را پوشش دهند. این امر به تیم‌های توسعه کمک می‌کند تا مطمئن شوید که نرم‌افزار به درستی کار می‌کند

**4کاهش هزینه‌ها:** در بلندمدت، تست خودکار می‌تواند هزینه‌های را کاهش دهد. این به دلیل کاهش زمان صرف شده برای تست‌های دسته‌ای و کاهش تعداد باگ‌های نرم‌افزاری است

**5افزایش بهره‌وری:** تست خودکار به تیم‌های توسعه اجازه می‌دهد تا خود را بر روی توسعه ویژگی‌های جدید بگذرانید. این امر باعث افزایش بهره‌وری تیم می‌شود

چالش های تست خودکار

1اولیه : پیاده‌سازی تست خودکار نیاز به سرمایه‌گذاری در ابزارها، آموزش و توسعه اسکریپت‌های تست دارد.

2عوامل فنی : رخی از انواع تست‌ها، مانند تست‌های کاربر پذیرنده، به سختی قابل خودکارسازی هستند.

**3 نگهداری اسکریپت‌های تست:** اسکریپت‌های تست نیاز به نگهداری و به‌روزرسانی دارند تا با تغییرات در نرم‌افزار سازگار شوند

ابزارهای تست خودکار

**سلنیوم:** یک ابزار محبوب برای تست خودکار وب‌اپلیکیشن‌ها است

**Appium :** یک ابزار برای تست خودکار اپلیکیشن های موبایل است

**Junit :** یک فریمورک تست برای زبان جاوا است.

**TestNG :** یک فریمورک تست برای زبان جاوا است.

**Robot Framework :** یک فریمورک تست بر کلمات کلیدی است.

Build Automation

**اتوماسیون ساخت** مجموعه‌ای از فناوری‌ها و روش‌هایی گفته می‌شود که فرآیندهای ساخت و ساز را خودکار می‌سازند. این هدف با هدف افزایش بهره‌وری، کاهش هزینه‌ها، بهبود کیفیت و ایمنی در پروژه‌های ساختمانی به کار می‌رود

چرا اتوماسیون ساخت دارد

**فزایش بهره برداری:** با حذف تکراری و دستی، استفاده از نیروی کار به طور قابل توجهی افزایش می‌یابد.

**کاهش‌های هزینه‌ها:** کاهش هزینه‌های اولیه طرح و زمان اجرای پروژه، استفاده از اصلی اتوماسیون ساخت هستند.

**بهبود کیفیت:** دقت و کیفیت ساخت با استفاده از ماشین آلات و ابزارهای مطمئن تضمینی می شود .

**فزایش ایمنی:** کاهش خطرات ناشی از کار و بهبود ایمنی از دیگر این فناوری ها است.

**سرعت بخشیدن به پروژه‌ها:** با حذف مراحل دستی و افزایش سرعت اجرای کارها، پروژه‌ها در زمان کوتاه‌تری به پایان می‌رسند.

**قابلیت‌پذیری بیشتر:** امکان ساخت سازه‌های پیچیده و سفارشی با استفاده از فناوری‌های اتوماسیون فراهم می‌شود.

فناوری های مورد استفاده در اتوماسیون ساخت

**ربات‌ها:** ربات‌ها در کارهای مانند جوشکاری، برش، رنگ آمیزی و حمل و نقل مواد به کار می‌روند.

**پرینترهای سه‌بعدی:** برای ساخت قطعات پیچیده و سفارشی و حتی ساخت کل سازه‌ها مورد استفاده قرار می‌گیرند

**درویدها:** دستگاه‌های خودکاری که قادر به انجام کارهای مختلف ساختمانی هستند.

**سنسورها:** برای جمع آوری داده ها در مورد محیط کار و وضعیت سازه مورد استفاده قرار می گیرند.

**نرم افزارهای طراحی و شبیه سازی:** برای طراحی و برنامه ریزی فرآیندهای ساختمانی استفاده می شوند.

کاربردهای اتوماسیون ساخت

**ساخت سازه های بتنی:** استفاده از ربات ها برای قالب بندی، آرماتوربندی و بتن ریزی.

**ساخت سازه های فولادی:** استفاده از ربات ها برای برش، جوشکاری و مونتاژ قطعات فولادی

**ساخت دیوارها و پارتیشن‌ها:** استفاده از پرینترهای سه‌بعدی برای ساخت دیوارهای بتنی یا گچ.

**تاسیسات مکانیکی و برقی:** استفاده از ربات‌ها برای نصب لوله‌ها، کابل‌ها و تجهیزات الکتریکی

**نگهداری و تعمیرا:** استفاده از پهپادها و ربات ها برای بازرسی و تعمیر سازه ها

چالش‌ها و آینده اتوماسیون ساخت

**هزینه‌های بالا:** فناوری سرمایه‌گذاری در اتوماسیون بالا است

**نیاز به نیروی کار متخصص:** کار با این فناوری نیازمند نیروی کار متخصص و آموزش دیده شده است.

**تغییرات در قوانین و مقررات:** قوانین قوانین و مقررات با فناوری‌های جدید چالش دیگری است.

**امنیت سایبری:** حفاظت از داده‌ها و سیستم‌های کنترل در برابر حملات سایبری بسیار زیاد است.

با وجود این چالش‌ها، آینده اتوماسیون ساخت بسیار روشن است. با پیشرفت فناوری و کاهش هزینه‌ها، انتظار می‌رود که اتوماسیون به طور خاص در صنعت ساخت و ساز مورد استفاده قرار گیرد. این امر منجر به افزایش بهره‌وری، کاهش هزینه‌ها و بهبود کیفیت ساختمان‌ها خواهد شد.

Release Automation

**اتوماسیون** (Release Automation) به مجموعه ای از ابزارها و فرآیندهایی گفته می شود که به صورت خودکار وظایف مربوط به انتشار نرم افزار، به روز رسانی ها و محتوا را انجام می دهد. این فرآیند شامل ساخت، تست، بسته بندی و استقرار نرم افزار در محیط های مختلف است. با اتوماسیون این فرآیندها، سازمان‌ها می‌توانند انتشار را افزایش دهند، کیفیت نرم‌افزار را بهبود بخشد و خطاهای انسانی را کاهش دهد

چرا اتوماسیون انتشار دارد

**افزایش سرعت انتشا:** انتشار اتوماسیون باعث می‌شود که نرم‌افزارها سریع‌تر به دست کاربران برسند.

**کاهش های انسانی:** با حذف انتشار انسان در بسیاری از مراحل، احتمال بروز خطا کاهش می‌ یابد

**بهبود کیفیت:** اتوماسیون تست و بررسی کیفی باعث می‌شود نرم‌افزار با کیفیت بالاتری منتشر شود

**فزایش بهره‌وری تیم:** تیم توسعه می‌تواند بر روی توسعه ویژگی‌های جدید انجام شود و از انجام کارهای تکراری رها شود.

**کاهش هزینه‌ها:** انتشار اتوماسیون باعث کاهش هزینه‌های مربوط به انتشار نرم‌افزار می‌شود.

اجزای اصلی اتوماسیون انتشار

**ساخت خودکار (Build Automation):** شامل کامپایل کد، اجرای تست‌ها و ایجاد بسته‌های قابل نصب است.

**ستقرار خودکار (Deployment Automation):** شامل انتقال بسته‌های نرم‌افزاری به محیط‌های مختلف (توسعه، تست، تولید) است

**مدیریت پیکربندی (Configuration Management):** شامل مدیریت و محیط‌های مختلف است

**یکپارچگی مداوم (Continuous Integration):** به صورت مداوم تغییرات کد را ادغام کرده و تست می‌کند.

**تحویل مداوم (تحویل مستمر):** به صورت مداوم نرم‌افزار را برای انتشار می‌کند.

ابزارهای رایج اتوماسیون انتشار

**Jenkins :** یک سرور CI/CD محبوب و قابل تنظیم است.

**GitLab CI/CD :** یک ابزار قدرتمند CI/CD که با GitLab یکپارچه شده است

**CircleCI :** یک پلتفرم CI/ CD مبتنی بر ابر است

**Ansible :** یک ابزار مدیریت پیکربندی و اتوماسیون است.

**Puppet :** یک ابزار مدیریت پیکربندی است.

**Chef :** یک ابزار مدیریت پیکربندی است

بهترین روش های اتوماسیون انتشار

**مدولار کردن اسکریپت‌ها:** کارهای کوچک را به بخش‌های کوچکتر و قابل استفاده مجدد تقسیم کنید

**استفاده از کنترل نسخه:** تغییرات در اسکریپت‌ها و پیکربندی‌ها را ردیابی کنید

**اتوماسیون تست:** تست‌های خودکار را به عنوان بخشی از روند انتشار بگنجانید.

**پیاده‌سازی CI/CD:** فرآیند ساخت، تست و استقرار را خودکار کنید

**ظارت و بهینه‌سازی:** عملکرد انتشار را به طور مداوم نظارت کرده و به دنبال راه‌های بهبود آن باشید.

Real-Time Software

**نرم افزارهای زمان واقعی (Real-Time Software** ) نوعی نرم افزار هستند که برای پاسخگویی به رویدادها در بازه های زمانی خاص طراحی شده اند. این نرم‌افزارها در سیستم‌هایی که نیاز به واکنش سریع و دقیق دارند، به کار می‌روند. سیستم‌های کنترل صنعتی تا دستگاه‌های پزشکی و خودروهای خودران، همه از این نرم‌افزارها استفاده می‌کنند

ویژگی های کلیدی نرم افزارهای زمان واقعی

**زمان‌بندی دقیق:** این نرم‌افزارها باید وظایف را در زمان‌های مشخص و با دقت بالا اجرا کنند.

**قابلیت اطمینان بالا:** این نرم‌افزارها باید بسیار قابل اعتماد باشند و در شرایط مختلف کار کنند

**کارایی بالا:** این نرم‌افزارها باید از منابع سیستم استفاده کنند.

**پاسخگویی سریع:** این نرم‌افزارها باید به سرعت به رویدادها پاسخ دهند.

انواع نرم افزارهای زمان واقعی

ن**رم‌افزارهای سخت زمان واقعی:** در این نوع نرم‌افزارها، عدم رعایت زمان‌بندی می‌تواند به عواقب جدی تبدیل شود. مثلاً در سیستم‌های کنترل پرواز، در پاسخگویی می‌تواند منجر به وقوع شود.

**نرم‌افزارهای نرم‌افزار زمان واقعی:** در این نوع نرم‌افزارها، عدم رعایت زمان‌بندی می‌تواند عملکرد سیستم را کاهش دهد، اما به طور کلی سیستم قابل استفاده خواهد بود. مثلاً در سیستم‌های چندرسانه‌ای، تأخیر در پخش صدا یا تصویر ممکن است آذردهنده باشد، اما سیستم همچنان کار می‌کند.

چالش‌های توسعه نرم‌افزارهای زمان واقعی

**زمان‌بندی‌های:** طراحی الگوریتم‌های زمان‌بندی کارآمد برای اطمینان از اجرای به موقع وظایف.

**مدیریت همزمانی:** مدیریت همزمان وظیفه و جلوگیری از شرایط شرایط مسابقه.

**مدیریت منابع:** بهینه سازی استفاده از منابع سیستم های مانند پردازنده، حافظه و دستگاه های ورودی/خروجی.

**قابلیت اطمینان:** طراحی سیستم‌هایی که در برابر خطاها مقاوم باشند و بتوانند از خود بازیابی کنند.

کاربردهای نرم افزارهای زمان واقعی

**صنعت:** کنترل فرآیندهای صنعتی، رباتیک

**خودرو:** سیستم های کنترل موتور، ترمز ABS

**پزشکی:** دستگاه های تصویربرداری، دستگاه های کمکی

**هوافضا:** سیستم‌های ناوبری، کنترل پرواز

**ارتباطات:** شبکه های مخابراتی، روترها

آینده نرم افزارهای زمان واقعی

* **هوش مصنوعی:** استفاده از هوش مصنوعی برای بهبود تصمیم گیری در سیستم‌های زمان واقعی
* **اینترنت اشیا:** توسعه نرم‌افزارهای زمان واقعی برای دستگاه‌های مرتبط با اینترنت
* **ابر محاسباتی:** استفاده از ابر برای اجرای نرم افزارهای زمان واقعی
* **امنیت سایبری:** افزایش امنیت نرمافزارهای زمان واقعی در برابر حرکات سایبری

**در کل، نرم افزارهای زمان واقعی نقش بسیار مهمی در زندگی روزمره ما ایفا می‌کنند. این نرم‌افزارها در بسیاری از دستگاه‌ها و سیستم‌هایی که با آن‌ها کار می‌کنند، وجود دارند و به طور مداوم در حال توسعه و بهبود هستند.**

-Embedded Software->BIOS->C, Rust

**نرم افزار تعبیه شده: قلب تپنده دستگاه ها**

نرم افزار تعبیه شده نوعی نرم افزار است که به طور اختصاصی برای کنترل سخت افزار یک دستگاه طراحی شده است. نرم‌افزارهای عمومی، این نرم‌افزارها در محدودیت‌های سخت‌افزاری دستگاه‌های کار می‌سازند و نیازمند پاسخگویی بلادر هستند.

**ویژگی های کلیدی نرم افزار تفسیر شده:**

* **زمانبندی دقیق:** نرم‌افزارهای باید در زمان‌بندی مشخص و با دقت بالا اجرا شود.
* **منابع منابع:** حافظه، قدرت پردازش و فضای ذخیره سازی محدود در اختیار دارد.
* **قابلیت اطمینان بالا:** باید بسیار قابل اعتماد باشد و در شرایط مختلف به خوبی کار کند.
* **بازدهی بالا:** باید از منابع سیستم به طور خودکار استفاده کرد.
* **پاسخگویی سریع:** باید به سرعت به رویدادها پاسخ دهد.

BIOS: پل ارتباطی بین سخت‌افزار و نرم‌افزار

BIOS (سیستم ورودی/خروجی پایه) یک قطعه نرم افزاری اساسی است که هنگام روشن شدن دستگاه، سخت افزارهای آن را مقداردهی اولیه می کند. وظایف اصلی BIOS عبارتند از:

* **خودآزمایی هنگام روشن شدن (پست):** بررسی می‌کند که همه اجزای سخت‌افزاری به درستی کار می‌کنند یا خیر.
* **بوت‌لودر:** سیستم عامل یا سایر نرم‌افزارها را می‌کند.
* **ورودی/خروجی پایه:** عملیات‌های ورودی/خروجی‌ها مانند کیبورد، ماوس و دیسک را مدیریت می‌کند.
* **زمان نگهداری:** زمان سیستم را حفظ می کند.

**ج: زبان غالب برای سیستم های تعبیه شده است**

C به مدت زبان اصلی برای توسعه نرم افزارهای تعبیه شده است. ویژگی اصلی C عبارتند از:

* **دسترسی مستقیم به سختافزار:** امکان کنترل دقیق بر روی سختافزار را می‌کند.
* **بازدهی بالا:** کد تولید شده توسط کامپایلر C بسیار بهینه است.
* **پشتیبانی** ابزارها و کتابخانه‌های بسیاری برای زبان وجود دارد.
* **زبان بالغ:** سال‌ها توسعه و بهبود یافته است.

**با این حال، C دارای برخی معایب نیز هست:**

* **پیچیدگی:** ساخت و نگهداری سیستم‌های پیچیده با C می‌تواند چالش‌برانگیز باشد.
* **مدیریت حافظه:** نیازمند مدیریت حافظه است که احتمال بروز خطا را می‌دهد.
* **آسیب‌های امنیتی:** در برابر امنیت مانند سریز بافر آسیب پذیر است.

Software Product Line

**خط تولید نرم‌افزار (خط تولید نرم‌افزار)** روشی منظم و برنامه‌ریزی شده برای توسعه خانواده‌های محصولات نرم‌افزاری است که بر اساس یک مشترک و مجموعه‌ای از ویژگی‌های قابل تغییر ساخته می‌شوند. این روش به سازمان‌ها می‌دهد تا محصولات نرم‌افزاری را با سرعت و کیفیت بالاتر تولید کند و در عین حال هزینه‌ها را کاهش دهد.

**خط تولید نرم افزار**

* **افزایش سرعت توسعه:** با استفاده از قطعات آماده و قابل استفاده، سرعت توسعه محصولات جدید به طور قابل توجهی افزایش می‌یابد.
* **کاهش هزینه ها:** کاهش هزینه های توسعه، نگهداری و پشتیبانی از محصولات نرم افزاری.
* **افزایش کیفیت:** با استفاده از روش های توسعه استاندارد و آزمایش های دقیق، کیفیت محصولات تضمین شده می شود.
* **افزایش توان‌پذیری:** قابلیت تطبیق سریع محصولات با نیازهای مشتریان و تغییرات بازار.
* **کاهش:** ریسک‌های مرتبط با توسعه از طریق استفاده از اجزای آزمایش شده و قابل اعتماد.

**جزء اصلی خط تولید نرم افزار**

* **محصول (Product Core):** مجموعه ای از ویژگی ها و اجزای مشترک در تمام محصولات خط تولید.
* **ویژگی‌های قابل تغییر (مدل‌های ویژه):** مجموعه‌ای از ویژگی‌های اختیاری که می‌توانند به محصولات مختلف اضافه شوند.
* **پلتفرم توسعه (سکوی توسعه):** ابزارها و فرآیندهای مورد نیاز برای توسعه، ساخت و تست محصولات.

**مراحل ایجاد خط تولید نرم افزار**

1. **تحلیل نیازمندی‌ها:** شناسایی محصولات مشترک و تولیدی.
2. **طراحی محصول هسته:** طراحی مشترک و ویژگی های قابل تغییر.
3. **ساخت پلتفرم توسعه:** ایجاد ابزارها و فرآیندهای مورد نیاز برای توسعه محصولات.
4. **توسعه محصولات:** ایجاد محصولات مختلف بر اساس محصول و ویژگی های قابل تغییر.
5. **نگهداری و توسعه:** نگهداری و بهبود خط تولید و محصولات موجود.

**کاربردهای خط تولید نرم افزار**

* **توسعه سیستم‌های مختلف:** تولید خانواده از سیستم‌های مختلف با ویژگی‌های مختلف.
* **توسعه نرم افزارهای سازمانی:** تولید محصولات نرم افزاری سازمانی با قابلیت شخصی سازی بالا.
* **توسعه نرم افزارهای موبایل:** تولید اپلیکیشن موبایل با ویژگی های مشترک و تغییرات.

**چالش های پیاده سازی خط تولید نرم افزار**

* **هزینه های اولیه:** هزینه های اولیه راه اندازی خط تولید می توان بالا باشد.
* **پیچیدگی مدیریت:** مدیریت خط تولید نیازمند دانش و مهارت های خاص است.
* **تغییرات در نیازمندی‌ها:** تغییرات در نیازمندی‌ها می‌تواند بر کل خط تولید عامل ایجاد شود.