شیوه تنظیم شماره نسخه در Sball

Sball Versioning Model

محدوده:

در این مستند نحوه تنظیم شماره نسخه (*Version*) برای کلیه ماژولهای اسبال شرح میگردد.

تاریخچه:

ردیف	نویسنده	تاریخ	شماره ویرایش	توضیحات
۱	تحقیق و توسعه فنی و مهندسی	۱۳۹۳/۱۰/۰۸	۱٫۰٫۱	
۲				

فهرست مندرجات

فهرست تصاویر

فهرست جداول

فهرست ضمائم

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

چکیده

نحوه تغییر شماره نسخه ماژولها نشانگر تغییر انجام شده و همچنین سازگاری با نسخه‌های پایینتر ماژول مورد نظر میباشد.

در این مستند در ابتدا شیوه تعریف شماره نسخه در *libtool* بررسی میبشود و در ادامه با توجه به الزامات برنامه‌های کلاستری، شیوه نسخه‌گذاری بسته‌ها و ماژولها مورد بررسی قرار میگیرد.

کلید واژه ها

version

۱ مقدمه

در مسیر توسعه محصول، استاندارد سازی شیوه تغییر شماره نسخه (*Version*) ماژولها بسیار مهم و ضروری میباشد. چرا که در فضای توسعه محصول پیگیری خط سیر تغییرات و همچنین نحوه سازگاری و یا ناسازگاری نسخه های مختلف ماژولها میتواند در فرآیند به روزرسانی و همچنین تولید ابزارهای مکمل بسیار مفید باشد.

از سوی دیگر تطابق شیوه اتخاذ شده با روشهای معمول در توسعه نرم افزارهای امری لازم است چرا که امکان تعامل با محصول برای دیگر توسعه دهندگان نیز راحت و قابل درک خواهد بود.

چنانچه محصول در یک فضای کلاستری در حال فعالیت باشد این موضوع میتواند اهمیت دوچندان داشته باشد جرا که قاعده نسخه گذاری عامل تعیین کننده ای در جهت سازگاری نسخه های مختلف محصول در یک کلاستر خواهد بود.

قاعده نسخه گذاری به مثابه شیوه آدرس دهی در مسیر توسعه محصول میباشد که علاوه بر هدایت مسیر توسعه، زبان واحدی جهت تعامل بین توسعه دهندگان ایجاد میکند.

آنچه در ادامه می‌آید شیوه‌ای جهت نسخه‌گذاری ماژولها و بسته ها در برنامه *Sball* به عنوان یک *Cluster*

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

Engine در یک فضای کلاستری میباشد. فضایی که نسخه‌های گوناگون برنامه در فضای کلاستر با یکدیگر در تعامل خواهند بود.

اسبال در محیط لینوکس توسعه یافته است و مفاهیم بکار رفته در قاعده نسخه گذاری تابع مفاهیم برنامه نویسی در سیستم عامل لینوکس میباشد.

۲ ساختار بسته‌بندی(*Packaging*)

هر بسته در اسبال به صورت یک *RPM* و یا *DEB* میباشد.

هر بسته میتواند مجموعه‌ای از ماژولها را در خود داشته باشد. هر ماژول به صورت یک کتابخانه (فایل *so*) میباشد.

۱.۲ ساختار شماره نسخه در بسته‌ها

شماره نسخه هر بسته دارای چهار عدد به شرح زیر است:

major.minor.micro-release

۲.۲ ساختار شماره نسخه در ماژولها

شماره نسخه ماژول مشخص کننده شماره نسخه فایل *so* خواهد بود که به این شرح است:

major.minor.micro

۳ شماره نسخه در *libtool*

فرآیند تولید بسته ها در توسط ابزارهای *Auto* همچون *Autoconf* و *Automake* خودکارسازی شده است. از آنجایی که کتابخانه‌ها توسط این ابزارها تولید میشوند لازم است شیوه نسخه گذاری در این ابزار را بدانیم.

شیوه تعریف شماره ویرایش برای کتابخانه فرضی *test* به این شرح است:

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

libtest_la_LDFLAGS= -version-info c:r:a

- *c*: نشانگر *Current* به معنای شماره ویرایش فعلی رابط (*Interface*) کتابخانه است.
- *r*: به معنای شماره *revision* یا شماره بازبینی است
- *a*: نشانگر *age* یا نسل است.

نحوه انجام تغییرات در اعداد از یک ویرایش به ویرایش دیگر به این شکل است:

- چنانچه کتابخانه نسبت به کتابخانه قبلی تغییری در رابط (*Interface*) ندارد و صرفا نحوه پیاده‌سازی تغییر کرده و یا باگی رفع شده است، عدد *r* یکی اضافه میشود.
- در غیر اینصورت در تمامی موارد *r* صفر میشود و عدد *c* یکی اضافه میشود و برای *a* دو احتمال وجود دارد
  - چنانچه رابط کتابخانه تغییر کرده است اما با ویرایش قبلی سازگار است (پشتیبانی میکند) عدد *a* یکی اضافه میشود.
  - چنانچه رابط کتابخانه به شکلی تغییر کرده که ویرایش قبلی را پشتیبانی نمیکند، عدد *a* به صفر تغییر میکند

با این فرمول، هنگامی که کتابخانه‌ای به صورت *c.r.a* ویرایش گذاری شود، بدین معنا است که تمامی *Interface*های از ویرایش *c-a* تا *c* را پشتیبانی میکند. کتابخانه حاصل از این ویرایش به این شکل نامگذاری خواهد شد:

libtest.so.$((c - a)).a.r

۱.۳ تلفیق با شیوه *major.minor.micro*

با توجه به اینکه عادتی در ویرایش گذاری برنامه‌ها با ترکیب سه عدد *major.minor.micro* وجود دارد، میتوان با فرمول زیر بین این دو استاندارد مطابقت ایجاد کرد:

$$c = major + minor$$

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

$r = micro$

$a = minor$

شروط حاکم بر نحوه تغییر این اعداد به این شکل است:

- چنانچه کتابخانه نسبت به کتابخانه قبلی تغییر رابط ندارد، و صرفا نحوه پیاده‌سازی تغییر کرده است و یا اینکه باگی رفع شده است، عدد *micro* یک عدد اضافه میشود.
- چنانچه رابط کتابخانه تغییر کرده است اما با ویرایش قبلی سازگاری دارد (عدم تغییر در پارامترهای ورودی و یا هر گونه تغییر ساختاری دیگر)، و صرفا مواردی به رابط اضافه شده است، عدد *minor* یک عدد اضافه میشود و عدد *micro* صفر میشود.
- چنانچه تغییرات در رابط کتابخانه با ویرایش قبلی سازگاری ندارد، عدد *major* یک عدد اضافه میشود و اعداد *micro* و *minor* صفر میشوند.

۴ نسخه‌گذاری در ماژولها

هر ماژول یک فایل *so* را تولید میکند.

۱.۴ قاعده ناسازگاری

هنگامی یک ماژول با نسخه قبلی خود ناسازگار خواهد بود که یکی از دو شرط زیر حادث شود(**قاعده ناسازگاری**) :

۱. تغییر در رابط کتابخانه شامل حذف رابط، تغییر در پارامترهای ورودی رابط.

✔ **نکته:** با این تعریف اضافه شدن رابط جدید، تا زمانیکه رابطهای قبلی حذف نشوند و یا ساختار ورودی آن‌ها تغییر نکند، ناسازگاری ایجاد نخواهد شد.

۲. ماژول جدید توانایی کار با ماژول نسخه قبل را، در کلاستر، نداشته باشد.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۲.۴ قاعده نسخه‌گذاری

بر اساس قاعده ناسازگاری، نسخه‌گذاری ماژولها به این شرح خواهد بود:

- به شرط وجود سازگاری و عدم تغییر رابط، عدد *micro* یک واحد اضافه میشود.
- به شرط وجود سازگاری و تغییر در رابط، عدد *minor* یک واحد اضافه میشود و عدد *micro* صفر میشود.
- در صورت وجود ناسازگاری، عدد *major* یک واحد اضافه میشود و اعداد *micro* و *minor* صفر میشوند.

۵ نسخه‌گذاری در بسته‌های

نسخه‌گذاری بر اساس تعداد ماژولهای موجود در بسته متفاوت است.

۱.۵ بسته دارای یک ماژول

در این بسته‌ها ، اعداد *major.minor.micro* دقیقا مطابق با شماره‌های متناظر در ماژول میباشند.

۲.۵ بسته دارای چند ماژول

نحوه تغییر اعداد *major.minor.micro* در اینگونه بسته‌های به این شرح است:

- در صورت تغییر *major* در هر کدام از ماژولها، عدد *major* بسته یک واحد اضافه شده و *micro* , *minro* بسته صفر میشوند.
- در صورت عدم تغییر در *major* و تغییر در *minor* هر کدام از ماژولها، عدد *minor* بسته یک واحد اضافه میشود و عدد *micro* صفر میشود.
- در صورت عدم تغییر در *major* و *minor* ماژولها و تغییر در *micro* هر کدام از ماژولها، عدد *micro* بسته یک واحد اضافه میشود.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۳.۵ قاعده عدد *revision*

عدد *revision* همیشه یک میباشد.

عدد *revision* تنها درصورتی تغییر خواهد یافت که بسته بر اساس تغییر در بسته های وابسته، بدون تغییر در شماره نسخه خود، نیاز به ساخت(*Compile*) مجدد داشته باشد.

تغییر در بسته های وابسته هنگامی موجب ساخت مجدد بسته مورد نظر ما خواهد شد که عدد *major* در آن‌ها تغییر کند.

در نتیجه در صورت تغییر در عدد *major* بسته های وابسته و عدم تغییر در شماره نسخه بسته جاری، بسته مجدد ساخته شده و عدد *revision* یک واحد افزایش می‌یابد.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com