Android init.rc service class

I'm not robot

reCAPTCHA

Continue

The Android Init language consists of five broad classes of operators: Actions, Teams, Services, Options and Imports. C-style backslash shoots can be used to insert white space into the marker. Double quotes can also be used to prevent text from breaking into multiple tokens. The rear lashes, when it is the last character on the line, can be used to fold the line. The lines that start with q (leading whitespace allowed) are comments. The system's properties can be enhanced with $property.name syntax. This also works in contexts where concats are required, such as import/init.recovery.$'ro.hardware.rc. Actions and services implicitly announce a new section. All commands or settings refer to a section recently announced. Teams or options before the first section are ignored. The services have unique names. If a second service is defined with the same name as the existing one, it is ignored and an error message is registered. Init .rc FilesThe init language is used in simple text files that accept the extension of the .rc file. There are usually several of them in several places in the system described below./system/etc/init/hw/init.rc is the main .rc file and the init downloads run at the beginning of its run. He is responsible for the initial setting up of the system. Init downloads all files contained in the directories /system,system_ext,vendor,odm,product/etc/init// immediately after downloading the primary /system/etc/init/hw/init.rc. This is explained in more detail in the Import section of this file. Outdated devices without a first-stage installation mechanism have previously been able to import init scripts for mount_all, however this is deprecated and not allowed for devices, Launch after q.The intention of these directories:/system/etc/init/ is intended for basic elements of the system, such as SurfaceFlinger, MediaService, and logd./vendor/etc/init/ is intended for soC-suppliers such as actions or daemons required for the basic functionality of SoC./odm/etc/init/ designed for device manufacturer items such as the device manufacturer' actions or activities. All services whose melon files are in the system, provider or odm sections must have their service records placed in the appropriate init .rc file located in the /etc/init/directory section where they are located. There is a macro build system LOCAL_INIT_RC that handles this for developers. Each init .rc file must additionally contain any activity related to its maintenance. An example would be userdebug/logcatd.rc and Android.mk files located in the system/core/registry directory. This LOCAL_INIT_RC in the logcatd.rc Android.mk file during the assembly process. Init downloads logcatd.rc mount_all commands and lets you run the service and stand in line when necessary. This gap of init .rc files according to their daemon is preferable to previously used monolithic init .rc files. This approach ensures that only service records that init reads and the only actions that init performs correspond to services whose holes are actually present in the file system, which was not the case with monolithic init .rc files. This will further help resolve merger conflicts by adding multiple services to the system, as each of them will go into a separate file. Actions are called command sequences. Actions have a trigger that is used to determine when an action is being performed. In an event that coincides with the trigger of the action, this action is added to the tail of the queue that will be executed (if it is already in the queue). Each action in the queue is dequeued in sequence, and each team in this action is executed in sequence. Init handles other activities (creating/destroying the device, setting up properties, restarting the process) between executing commands in actions. Actions take shape: Actions are added to the queue and performed on the basis of the file review order, which its zlt'trigger'gt; zlt'trigger'g\t; contains a zlt'gt; zlt'command'gt; zlt'command'gt; (see import section) and then sequentially in a separate file. For example, if the file contains:on boot setprop a 1 setprop b 2 on the download property: true'true setprop c 1 setprop d 2 on the download setprop e 1 setprop f 2 Then, when the download trigger occurs and assuming that the property is truly equally true, then the order of the executed commands will be: setprop 1 setprop b 2 setprop c 1 setprop d 2 setprop e 1 setprop f 2 ServicesServices - these are programs that are thrust and (optional) restarted at the exit. Services take the form of: zlt'gt; OptionsOptions are modifiers for services. They affect how and when init launches the service and Install opportunities with exec'ing uth since. Opportunity should be a Linux feature without a prefix of CAP_ like NET_ADMIN or SETPCAP. See for The Linux Feature List. If opportunities are not provided, all options are removed from the service, even if it works as root.class and Indicate class names for the service. All services in the class may be run or stopped together. The service is in the default class if it is not listed through the class option. Additional class names that go beyond (necessary) first are used for group services. The animation class should include all the services required for both download animation and off animation. эти службы могут быть запущены очень рано во время загрузки и могут работать до последнего этапа остановки, доступ к /разделу данных не</name> </name>

</capability></argument></pathname></name></command></command></command></trigger></trigger> </capability></argument></pathname></name></command></command></command></trigger></trigger> These services can check files under/data, but it should not keep files open and should work when /data'lt'lt'gt;not available. An additional second option chooses a specific console instead of the default. By default, the kernel /dev/console can be changed by setting the androidboot.console kernel. In all cases, the presenters /dev/ must be omitted, so /dev/tty0 will be listed as simply a tty0 console. This option connects stdin, thick, and stderr to console. It is mutually exclusive with stdio_to_kmsg that connects only thick and stderr with kmsg.criticalthe device critical services. If it goes out more than four times in four minutes or before the download is completed, the device will restart in bootloader.disabledThis service will not automatically start with its class. It should be explicitly launched by name or by interface name.enter_namespace the zlt'gt; zlt'gt; introduces a type name space on the way. Only name network spaces are supported by a set of types for the network. Please note that only one name space of this type can be entered. For natives executed see libcutils android_get_control_file ().Group Change on the groupname'gt; groupname'gt'gt'group name before exec'ing of this service. Additional group names that go beyond (necessary) first are used to set up additional process groups (through setgroups). It is now the default for root. (??? probably should't default to anyone) As far as Android M, processes should use this option, even if they require Linux capabilities. The process had to work as a root, request opportunities, and then fall to the desired uid. There is a new mechanism through the fs_config system that allows device interface name should be a fully qualified name, not a valued name. For example, it is used to allow a servicemanager or hwservicemanager to lazily start a service. This tag should be used several times when using multiple interfaces. An example of a recording for the HIDL interface is the vendor.foo.bar@1.0::IBaz default interface. For the AIDL interface, use the aidl interface. The name of the instance for the AIDL interface is what is registered in the servicemanager, and they can be listed with adb shell dumpsys -I.ioprio Sets IO priority and IO priority class for this service through class SYS_ioprio_set syscall. the class should be one of the zlt'gt; The priority should be in the range of 0 - qlt'keycode. If all the keys that match the passed key codes are pressed at once, the service will start. This is commonly used to run an error reporting service. This option can take the property instead of a list of key codes. In this case, only one option is available: the name of the property in a typical расширения свойства. Свойство</keycode> </keycode></priority&gt_alt:class> </instance> </instance> </interface> </groupname> </groupname> </type> </path> </seclabel> </group> </user> </perm> </max> </cur> </resource> </resource>, the same.property.name equals to 123,124,125. Since key codes are processed very early init, only PRODUCT_DEFAULT_PROPERTY_OVERRIDES properties can be used.memcg.limit_in_bytes and memcg.limit_percent Sets a child memory.limit_in_bytes of the value and limit_percent which is interpreted as a percentage of the device's physical memory size (only if memcg is installed). Values must be equal or 0.memcg.limit_property that set a child's memory.limit_in_bytes to the value of the specified property (only if memcg is installed). This property will override the values listed through the memcg.limit_in_bytes and memcg.limit_percent.memcg.soft-limit-in'bytes Sets the child's memory.soft_limit_in_bytes to the specified value (only if memcg is set), which should be equal or 0.mecg.swappiness Sets the child's memory.swappiness to specified value (only if memcg is installed), which should be equal to or more than 0.space name. Enter a new PID or mount a name space when forking service.oneshotDo don't restart the service, when it comes out.onrestartExecute command (see below) when the service restarts.oom_score_adj to a specified value, which should range from -1000 to 1000.overrideIndicates that this definition of the service is intended to override the previous definition for the service of the same name. This is especially designed for services on /odm to override those that are identified by/supplier. The last definition of a service that init analyzes with this keyword is the definition the service will use for that service. Pay close attention to the init.rc file review because it has some features for backward compatibility reasons. In the import section of this file, more information about the priority of the order. This should range from -20 to 19. The default priority is 0. Priority is set through setpriority () .reboot_on_failure If this process cannot be started or if the process is completed with a release code other than CLD_EXITED or status other than '0', restart the system with the target target. This is especially designed to be used with exec_start built in for any must-have check during the boot.restart_period If the non-oneshot service comes out, it will be restarted at its start time plus this period. It defaults to 5s to estimate the service failure limit. This is быть увеличено для служб, которые предназначены для периодического запуска. Например, он может быть установлен на 3600, чтобы указать, что служба должна</seconds> </target> </priority> </value> </pid|mnt> </value> </value> an hour or 86400 to indicate that the service must work every day.rlimit This is a zlt.gt; zlt'gt; qgt'gt; qlt'gt;applies this rlimit to the service. rlimits are inherited by children's processes, so this effectively applies this rlimit to the process started by this service. It is analyzed in the same way as the setrlimit team listed below. Primarily for use by services run from the root, such as uevendb. adbd. Services in the system section may instead use certain policy transitions based on the file security context. If not specified and the transition is not defined in the policy, default init context.setenv Set a variable environment name for the value in the zlt'gt; the running process.shutdown shutdown_behavior.shutdown_behavior.shutdown_behavior.shutdown_behavior.install the behavior of shutting down the service process. If this is not specified, the service dies during the stop process with SIGTERM and SIGKILL. Service with critical shutdown_behavior do not die during a stop before switching off. When time is cut off, even services marked by a critical shutdown will be killed. When a service flagged for critical completion does not start when it is disconnected, it will be launched now. Typically, it is used when a variable loader is set, indicating that a particular service should be started if necessary. For example, on a property:ro.boot.m/fancyhardware-1, include my_fancy_service_for_my_fancy_hardware exec zlt;seclabel. The team starts after --, so that additional security context, user and additional groups can be provided. No other commands will run until this one is complete. seclabel can be - to refer to the default. Properties are expanded within <name> </group> </seclabel> . SO_PASSCRED . The user and group are up to 0 by default. 'seclabel' is the S security context for the outlet. By default, it is performed in the context of the security of the service, as specified by seclabel or calculated based on the security context of the service file. For native executables see libcutils android_get_control_socket () .stdio_to_kmsgRedirect thick and stderr to /dev/kmsg_debug. This is useful for services that don't use native Android registration during early downloads and whose message logs we want to capture. This is only included when /dev/kmsg_debug included, which is only included on userdebug and eng builds. This is mutually exclusive with the console option, which additionally connects stdin with the data console.task_profiles and Set profiles of the zlt'gt; zlt'gt;tasks for the process when it plugs. This is designed to replace the use of the option для перемещения процесса в group.timeout_period </seconds></seconds></seconds>обеспечит тайм-аут, после которого служба будет убита. Oneshot ключевое слово соблюдается здесь, так oneshot услуги не автоматически перезапустить, однако все другие услуги будут. Это особенно полезно для создания периодической службы в сочетании с restart_period вариантом</seconds> </profile> </profile> </type> </perm> </type> </name> </seclabel> </value> </name> </seclabel> </group> </user> </perm> </max> </cur> </resource> </resource> that the service can be override (through the override option) later in the APEXes download sequence. When the updatable service is up and running before APEXes is activated, the execution is delayed until the activation is complete. A service that is not labeled as updatable cannot be redefined by APEXes.user zlt;username'gt;Change to the user's name until exec'ing of this service. It is now the default for root. (???, probably shouldn't default to anyone) Previously, to acquire Linux capabilities, the process had to work as a root, request opportunities, and then fall to the desired uid. There is a new mechanism through the fs_config system that allows device manufacturers to add Linux capabilities to specific movie deacons in the file system that should be used instead. This mechanism is described in . When using this new mechanism, processes can use the custom option to choose the desired uid without even working as a root. As that as Android O, processes can also request opportunities directly in their .rc files. See the option below.writepid - qlt'zlt'gt; write a child's pid in these files when he plugs. Designed to use cgroup/cpuset. If the files under /dev/cpuset/ are not specified, but the 'ro.cpuset.default' feature is set up to the non-empty name cpuset (e.g.,/foreground), then pid is written for the /dev/cpuset/cpuset_name/tasks file. Using this option to move the process to a group is out of date. Please use task_profiles instead. TriggersTriggers are strings that can be used for matching events and are used to occur. Triggers are divided into event triggers and property triggers. Event triggers are lines caused by a trigger or queueEventTrigger function within the irrefutable range. They take the form of a simple line, such as downloading or late init. Property triggers are strings called when a named property changes value to a given new value, or when a named property changes value to any new value. They take the form of property: and property: accordingly. Property triggers are further evaluated and run accordingly during the initial init loading phase. The action may have multiple property triggers, but can only have one event trigger. For example: on a download property: a'b'lt;'file'gt; zlt;zlt;'zlt;'s qgt; defines an action that occurs only when a download event trigger occurs, and the property is equal to a b.on.a'b'd defines an action that is performed three times: During the initial if the property is a'b and the property c'd. Any time that property goes to value b, while property c is already equal to d.Any time that property c goes to the value d, while the property is already equal to value (start) stop-start/stop butcharring. They are present in init.rc files by default, but but bootcharring is only active if the file /data/bootchart/enabled exists; otherwise, bootchart start/stop are permissions to access the files no-ops.chmod and group.class_start qrt'gt;,For more information on the launch of services.class_start_post_data Like class_start, you can only get information about the launch of the services.class_start_post_data Like class_start, but only take into account the services that were launched after the installation /data and which were launched at the time when the class_reset_post_data was called. Used only for FDE devices.class_stop and disables all services of this class if they currently running.class_reset Stop all of the services of the specified zlt.s. ar) Like writing, but useful for binary/large amounts of data. As for the src file, copying from the symbolic link file and world-writable or group-writable files are not allowed. As for the dst file, the default mode is 0600 if it doesn't exist. And it will be truncated if the dst file is a normal regular file and already exists.domainname zlt'gt;Install a domain name. Enable is turned the disabled service into a enabled service as if the service hasn't been disconnected. If the service is expected to be launched, it will be launched now. Typically, it is used when a variable loader is set, indicating that a particular service should be started if necessary. For example, on a property:ro.boot.m/fancyhardware-1, include my_fancy_service_for_my_fancy_hardware exec zlt;seclabel. The team starts after --, so that additional security context, user and additional groups can be provided. No other commands will run until this one is complete. seclabel can be - to refer to the default. Properties are expanded within the argument. Init stops the execution of commands until the split process exits.exec_background and executes a team with the zlt.seclabel'gt; This is handled in a similar way to the Exec team. The difference is that the init is not выполнения команд до тех пор, пока процесс не выйдет для тех пор, пока процесс не выйдет для тех пор, пока процесс не выйдет для данной службы и не остановит обработку дополнительных команд init до тех пор, пока она не вернется. Команда функционирует аналогично команде exec, но используют</seclabel> </argument> </command> </group> </user> </argument> </command> </group> </user> </seclabel> </servicename> </name> </dst> </src> </serviceclass> </serviceclass> </serviceclass> </serviceclass> </path> </argument> </path> </octal-mode> </octal-mode> Definition of the service instead of the exec vector.export argument Install a variable medium name, equal value in a global environment (which will be inherited by all the processes initiated after the implementation of this zlt;name)Hostname Set the name of the host. -f: The propulsion system of the module, even if the running core version and the kernel version, for the zlt'name, for the zlt'name, the name of the interface, which was designed to be a module, does not match.interface_start interface_restart interface_stop find a service that provides the interface name if it exists and run a start, or stop command on it accordingly. , or the name AIDL, in which case it is listed as aidl/qlt;interface/interface for example android.hardware.secure_element@1.1::ISecureElement/eSE1 will launch HIDL, which provides a copy of android.hardware.secure_element@1.1 and eSI1. interface_start/aidl_lazy_test_1 will launch an AIDL operate on interfaces defined by the interface option, not on interfaces registered during execution. Example of using these commands: interface_start android.hardware.secure_element@1.1::ISecureElement/eSE1 will launch HIDL, which provides a copy of android.hardware.secure_element@1.1 and eSI1. interface_start aidl/aidl_lazy_test_1 will launch an AIDL service that provides aidl_lazy_test_1 interface.load_system_props (this action is deprecated and non-op.) load_persist_propsLoads properties when/data has been decrypted. This is included in the default init.rc.loglevel Sets the level of the init.rc.loglevel, for 7 (all magazines) to 0 (fatal entry only). The numerical values correspond to the levels of the kernel log, but this command does not affect the level of the kernel log. Use the writing command to write /proc/sys/kernel/printk to change that. Properties are expanded within level.mark_post_dataUsed to mark the point immediately after/data is set. Used to implement class_reset_post_data and class_start_post_data.mkdir. Provided the mode, the owner and the group will be updated if the directory already exists. the directory will be encrypted if the parent. Required: encrypt the directory, interrupt the download process if encryption fails: try to establish an encryption policy, but continue if it doesn't remove the catalog if necessary to set the policy.key encryption can be one of: ref: use a system DE keyper_boot_ref: use a key freshly generated on </action> </name> </group> </owner> </mode> </path> </interface> </instance> </interface> </path> </interface> </owner> </path> </octal-mode> </path>] The challenges is_mgr_swapon_all this fs_mgr fstab with additional early and late options. With an early set, the init performed will skip the mount-up of records with the latemount flag and launch the FS encryption state event. With a set --late initiabe will mount records only with the flag latemount. The default option is not set, and mount_all will process all records in this fstab. If fstab is not specified, fstab.$ro.hardware fstab.$ro.hardware or fstab.$ro.hardware.platform _flag_s) Options include barrier No.1, noauto_da_alloc, opt out, ... as a comma of a separated line, such as a 1.noauto_da_allocperform_apex_configPerforms task after APEXes is installed. For example, it creates data catalogs for APEX configurations, analyzes the file configuration configuration configurations of configuration configuration configurations of configuration configuration configurations of configuration configuration configurations of configuration configuration configurations of configuration configurations of configuration configuration configuration configuration configurations of configuration configurations of configuration configuration configurations of configuration configuration configurations of configurations of configurations of configurations of configurations of configuration configurations of configuration configurations. Designed to be used only once, apexd notifies a fixing event by setting apexd.status to ready.restart.stops and restarts the running service. It drops nothing if the service currently restarts, otherwise it simply launches service.restorecon - Restore a file called the file_contexts'It'pathgt; on the way to the It. Not required for directories created by init.rc, as they are automatically marked correctly init.restorecon_recursive - Re-restore the catalog tree based on the way to the contexts of the It'gt) security, specified in the configuration of the file_contexts.rm't'pathgt; on the way to the It. You can use exec - rm ... Instead (assuming that the section of the system is already installed).rmdir qlt'gt'gt;calls rmdir (2) on this path.readahead (file)dir'gt; (--completely)Calls readahead (2) on the file or files in this catalog. Use the --full option to read the full name of the content.setprop.gt.' Properties have been expanded within value.setrlimit.gt; This applies to all processes that run after the limit. It is designed for early init and is applied all over the world. the resource is best indicated by text view ('cpu', 'rtio', etc. or 'RLIM_CPU', 'RLIM_RTIO', etc.). It can also be listed as an int value, which enum ресурса. Обратите внимание, что это не синхронно, и даже если бы это было так, нет никакой гарантии, что планирующая операционной системы будет выполнять</service> </max> </value> </resource> </service> </file)dir> </path> </path> </path> </service> </options> </fstab> </dir> </device> </type> </fstab> </type> </fstab> enough to guarantee anything about the status of the service. See exec_start commands for the synchronized version of the launch. This creates an important consequence that if a service offers functionality to other services, such as providing a communication channel, simply running the service before those services are not enough to ensure that the channel was created before those services ask for it. There needs to be a separate mechanism to make any such guarantees.stop the service from running if it is currently running.swapon_all - fs_mgr_swapon_all in this fstab file. If the fstab option is not specified, fstab.$ro.boot.fstab_suffix, fstab.$ro.hardware or fstab.$ro.hardware platform) will be scanned for under /odm/etc, /vendor/etc, or/during execution. In the It;gt order:gt;order.symlink Create minutes_west_of_gmt a symbolic link on the way to the target values of the It;event); Used for a queue of actions from another action.umount q'tr'gt'unmount file system, installed in the path.unmount_all - fs_mgr_unmount_all qlt.fstab'gt;. If the fstab option is not specified, fstab.$ro.boot.fstab_suffix, fstab.$ro.hardware or fstab.$ro.hardware.platform will be scanned for under /odm/etc, /vendor/etc, or /at runtime, is that order.verity_update_state the Inner implementation detail used to update the state of dm-verity and install the zlt'mount-point'gt.partition.mount-point.verified properties used by dm-verity remount, because fs_mgr can't install them qlt;path'gt;directly.qlt.timeout'gt;self. If the timeout is not specified, it is currently defaulted to five seconds. The value of a timeout can be fractional seconds at the floating point notation.wait_for_prop to wait until the name of the system's property is valued. Properties are expanded within the cost range. If the property name is already set up for the value, continue immediately.write.It'gt; zlt'gt;Open the file on the way and write a line to it with write (2). If the file does not exist, it will be created. If it exists, it will be truncated. Properties are expanded in content. Importsimport is an init config file that expands the current configuration. If the path is a directory, each file in the catalog is analyzed as a config file. This is not recursive, the nested catalogs will not be dismantled. The import keyword is not a command, but rather a command of its own section, which means that this does not happen as part of the action, but rather the import is processed as the file is parsing and following the following logic. There are only three times when the init is

executed importing .rc files: When it imports /systems/etc./init/hw/init.rc or the scenario specified by the property ro.boot.init_rc during the initial download. </content> </path> </value> </name> </timeout> </path> </mount-point> </fstab> </path> </event> </minutes_west_of_gmt> </path> </target> </fstab> </service> </service> imports/System, system_ext,supplier,odm,product/etc/init/ immediately after import /system/etc/init/hw/init.rc. When it imports /system_supplier, odm/etc/init/ or .rc files on certain paths for mount_all years, is not allowed for devices launched after the order that files are imported a little difficult for outdated reasons. The following is guaranteed:/system/etc/init/hw/init.rc is then re-analyzed each of its imports. Content /system/etc/init) is analyzed in alphabetical order and analyzed sequentially, with imports recursively after each file is disassembled. Step 2 is repeated for /system_ext/etc/init, /vendor/etc/init, /odm/etc/init, /product/etc/initThe below pseudocode may explain this more clearly:fn Import (file) Parse (file) for (import: file.imports) Imports (/system/etc./init/hw/init.rc) Catalogs /system_ext/etc/init /odm/etc/init, /product/etc/init) for (catalogue: Catalogs) files for (file: files) Imports (file) Actions are performed in the alphabetical order of the directory's contents, if they are disassembled. For example, post-data (s) actions in /system/etc/init/hw/init.rc are always the first action after receiving the data (s) that must be performed in order to run it in that file. Then import actions /system/etc/init/hw/init.rc in order of their import, etc. PropertiesInit provides government information with the following properties.init.svc. The state of the service (stopped, stopped, stopped, running, reboot) dev.mnt.blk. Mount_point the basic name of the device block associated with mount_point. Mount_point has/replace. and if you refer to the root attachment point/, it will use /root, in particular dev.mnt.blk.root. Designed to refer to /sys/device/block/$'dev.mnt.blk.'lt;mount_point/and /sys/fs/ext4/$'dev.mnt.blk.'lt;mount_point/adjust the device's block characteristics in an agnostic way. Init reacts to properties that start with ctl. These properties adopt the ctl format. The system's properties are used as a parameter. The goal is not mandatory and determines the service option by which the value should be matched. There is only one option for the target, the interface, which indicates that the value will relate to the interface that the service provides, not the name of the service itself. For example: SetProperty (ctl.start, logd) will launch a launch team on the log. SetProperty (ctl.interface_start, aidl/aidl_lazy_test_1) will launch a launch team on a service that provides helpl aidl_lazy_test_1 interface. Note that these properties are only meshable; they won't matter when viewed. The teams are listed below. Start rebooting stop эквивалентны использование команды запуска, перезагрузки и остановки на службе, указанной значением property.oneshot_on</command> </target> </mount_point> </mount_point> </name> </Alphabetical> </Alphabetical> oneshot_off will turn the oneshot flag on or off for the service specified in the property value. This is especially designed for services that are conventionally lazy HALs. When they are lazy HALs, one short should be on, otherwise the oneshot has to be off.sigstop_on and sigstop_off will turn on or off the sigstop function for the service specified in the property value. For more information on this feature, see Debugging init below. Boot timingInit records some information about download times in the system properties.ro.boottime.initTime after downloading to ns (through the hours CLOCK_BOOTTIME), on which the first stage of the init started.ro.boottime.init.first_stageHow long in ns it took to run the first stage.ro.boottime.init.selinux How long in ns it took to run SELinux stage.ro.boottime.init.cold_boot_waitHow long init waited 19th time. The time after downloading to ns (through hours CLOCK_BOOTTIME) that the service was first launched. BootchartingThis version of init contains code to perform download: generating log files that can later be processed by tools provided by the emulator use the timeout option to download with a download load activated for the seconds of timeout. On device: adb shell 'touch /data/bootchart/enabled' Be sure to delete this file when you have finished collecting data! The log files are written on /data/bootchart/. The script is provided to retrieve them and create a bootchart.tgz file that can be used with the bootchart command line utility:sudo apt-get install pybootchartgui and grab-bootchart.sh uses $ANDROID-SERIAL. $ANDROID-BUILD-TOP/system/core/init/grab-bootchart.sh One thing to watch for is that bootchart will show init if it started running at 0s. You'll have to look at dmesg to work when the core actually started init.Comparing two bootchartsA handy script called compare-bootchart.py can be used to compare the start/end of selected processes. The aforementioned grab-bootchart.sh leave the butchart tarbol named bootchart.tgz on/tmp/android-bootchart. If two of these tarballs are stored on the host machine under different catalogs, the script can list the time differences. For example: Use: system/core/init/compare-bootcharts.py base-bootchart-dir exp-bootchart-dirprocess: basic experiment (delta) - unit ms (jiffy is 10ms on the system) -----------------------------------/init: 50 40 (-10) /system/bin/surfaceflinger: 4320 4470 (No150) /system/bin/bootanimation: 6980 6990 (No10) zygo 10410 10640 (No230) Sigota: 10410 10640 (No 230) system_server: 15350 15150 (-2 system_server: 15350 15150 (--- 200) bootanimation ends at: 33790 31230 (-2560) SystraceSystrace Can be used to obtain performance analysis reports while loading time on userdebug or eng builds. Here's an example of a track of the zlt;service-name Wm and am categories: $ANDROID-BUILD-TOP/external/chromium-trace/systrace.py and wm am-boot This command will result in a reboot of the device. After rebooting the device and completing the download sequence, the track report is received from the device and written as trace.html on the host, clicking on Ctrl-C.Limitation: The tracing events record are triggered after the persistent properties are loaded, so the trace events that have been emitted before are not recorded. Some services, such as vold, surfaceflinger and servicemanager, suffer from this restriction, as they are launched before the loading of persistent properties. The initialization of the zigota and the processes forked out of the zigota are not affected. Debugging init When the service starts with an init, it may not exec () the service. This is not typical, and may indicate an error is happening in the linker as the new service is running. The binder in Android prints its logs to enter and stderr, so they are visible in the logcat. If an error has been made before you can access the log cat, the stdio_to_kmsg service option can be used to overtake the logs that the binder prints, on the stderr to kmsg, where they can be read through a serial port. Running init services without init is not recommended because init creates a significant amount of environment (user, group, security tag, features, etc.) that are difficult to reproduce manually. If you want to debug the service from the start, you add a sigstop service option. This option will send SIGSTOP to the service just before the exec call. This gives a window where developers can attach a snooger, strace, etc. before continuing the service with SIGCONT. This flag can also be dynamically controlled by ctl.sigstop_on and ctl.sigstop_off properties. Below is an example of the dynamic debugging that is entered through the above: stop logging in the setprop ctl.sigstop_on log ps-e grep log to enter qgt; 4343 1 18156 1684 do_signal_stop 538280 T init gdbclient.py -p 4343 b main c c qgt; breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, breakpoint 1, (argc1, a.m. argv'0x7ff8c9a488) in the system/core/logd/main.cpp:427 Below is an example of doing the same, but with stracestop prepared setprop ctl.sigstop_on the blank in the journal of the start of registration ps-e grep registered qgt; 434 registered 3 1 18156 1684 do_signal_stop 538280 T init strace -p 4343 (From the other shell) kill -SIGCONT 4343 qgt; strace runs Host Init Script VerificationInit scripts are tested for correctness during assembly time. In particular, the following is checked. Well-formatted activities, services, and imports, such as no action without the previous on line and without any foreign lines after the import application. All command cards on the actual keyword and the number of arguments is in the right range. All service options are valid. It's stricter than how teams are tested, because service options are completely disassembled, for example, must be decided by UID and GID. There are other parts of init init which are only parried during the run and therefore not checked during the assembly, among them below. The validity of the commands' arguments, such as the lack of verification, whether the file paths actually exist, whether SELinux authorizes the operation, or if UIDs and GIDs allow it. There is no verification of whether the service exists or has a valid SELinux definedNo verification domain if the service has not previously been defined in another init script. Early init Boot SequenceThe early init loading sequence is broken down into three stages: the first phase of the init, the SELinux installation, and the second stage of the init. The first phase of init is responsible for setting minimum requirements for downloading the rest of the system. Specifically, this includes mounting/dev, /proc, mounting 'early mount' sections (which should include all sections that contain system code, such as a system and vendor), and moving system.img-mountain/for devices with ramdisk. Note that in Android, system.img is always TARGET_ROOT_OUT and always set on/by the time the first phase of the init ends. Android will also require dynamic sections and therefore will require the use of ramdisk to download Android. Restoring ramdisk can be used to download on Android rather than dedicated ramdisk as well. The first phase of init has three variations depending on the configuration of the device: For system-like root devices, the first phase of init is part /system/bin/init and symlink at/init points to /system/bin/init for backward compatibility. These devices don't need to do anything to mount system.img, since it is by definition already installed as rootfs core. For devices with ramdisk, the first phase of the init is a static performed located on /init. These devices mount system.img-like/system then perform a switch root operation to move the attachment to /system to/. The ramdisk content is released after the installation is completed. For devices that use recovery as a ramdisk, the first phase of init it is contained in the total init located on/init within the ramdisk recovery. These devices first switch the root to /first_stage_ramdisk to remove recovery components from the environment, and then continue the same as 2). Please note that the decision to download to Android instead of downloading to recovery mode is made if the androidboot.force_normal_boot'1 is present in the command line of the kernel. After the first stage the init finishes it execs /system/bin/init with an argument selinux_setup argument. At this point, SELinux is additionally compiled and uploaded to the system. selinux.cpp provides more information about the specifics of this process. Finally, once this stage ends, it execs /system/bin/init again with an argument second_stage argument. At the moment, the main phase of init is triggering and downloading through init.rc scripts. Scenarios. Scenarios. android init.rc service class late_start