RK３３９９开发遇到的问题

1、Ubuntu在安装一系列库的时候遇到的问题

**Ubuntu 16.04安装MinGW32**

2016年12月04日 23:47:14

阅读数：2632

直接使用命令

sudo apt-get install mingw32

会报错

Unable to locate package mingw32

解决办法如下

sudo gedit /etc/apt/sources.list

//在sources.list末尾添加，保存

deb http://us.archive.ubuntu.com/ubuntu trusty main universe

再次执行

sudo apt-get update

sudo apt-get install mingw32

会报下面的错误：

**E: 无法获得锁 /var/lib/apt/lists/lock - open (11: Resource temporarily unavailable)解决方法**

2014年09月19日 02:29:03

阅读数：15762

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Author  : Samson  
 \* Date    : 09/20/2014  
 \* Test platform:  
 \*              Linux ubuntu 3.2.0-58-generic-pae  
 \*              GNU bash, version 4.2.39  
 \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

在进行apt-get update操作时，出现如下的错误信息：

**v0idp0int-eof # apt-get update   
E: 无法获得锁 /var/lib/apt/lists/lock - open (11: Resource temporarily unavailable)  
E: 无法对目录 /var/lib/apt/lists/ 加锁**

这个问题是的意思是用于apt-get操作时的lock(应该是用于互斥的锁)，出现了不能够得到锁的问题，针对此问题，如何解决呢？

可以以下命令进行操作，  
sudo rm /var/lib/apt/lists/\* -vf

sudo apt-get update

然后，再次进行apt-get update操作，应该就可以了。

**ubuntu16.04 更新安装软件出错修复**

2016年06月26日 17:39:04

阅读数：8085

Get:1 http://211.162.127.1/files/3160000000353389/mirrors.163.com/debian xenial/main amd64 libxext-dev amd64 2:1.3.3-1 [82.1 kB]  
Err:1 http://211.162.127.1/files/3160000000353389/mirrors.163.com/debian xenial/main amd64 libxext-dev amd64 2:1.3.3-1  
  Writing more data than expected (82816 > 82090)

E: Failed to fetch http://211.162.127.1/files/3160000000353389/mirrors.163.com/debian/pool/main/libx/libxext/libxext-dev\_1.3.3-1\_amd64.deb  Writing more data than expected (82816 > 82090)

E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?

Unable to locate package错误，等等

解决：

1.sudo apt-get updata

2.sudo apt-get install XXX  --fix-missing

如果还不行可以采用以下方式

I'm running Linux Mint 18 (Sarah) and today I got the following update error:

# sudo apt update

Ign:1 http://ftp.nluug.nl/os/Linux/distr/linuxmint/packages sarah InRelease

...

Err:12 http://mirrors.nic.funet.fi/ubuntu xenial-updates/main amd64 Packages

Writing more data than expected (863064 > 862917)

Ign:13 http://mirrors.nic.funet.fi/ubuntu xenial-updates/main i386 Packages

Fetched 204 kB in 3s (53.7 kB/s)

Reading package lists... Done

E: Failed to fetch http://mirrors.nic.funet.fi/ubuntu/dists/xenial-updates/main/binary-amd64/Packages Writing more data than expected (863064 > 862917)

E: Some index files failed to download. They have been ignored, or old ones used instead.

However, the old school solution as shown in various (duplicate) posts is not working and without any effect. (I.e. same output as above.)

sudo apt -o Acquire::https::No-Cache=True -o Acquire::http::No-Cache=True update

安装交叉编译环境时出错：

sudo apt-get install gcc-arm-linux-gnueabihf \

gcc-aarch64-linux-gnu device-tree-compiler lzop libncurses5-dev \

libssl1.0.0 libssl-dev

The following packages have unmet dependencies:

gcc-arm-linux-gnueabihf : Depends: gcc-5-arm-linux-gnueabihf (>= 5.3.1-3~)

E: Unable to correct problems, you have held broken packages.

解决方法：先安装 gcc-arm-linux-gnueabihf 在安装其他的  
sudo apt-get install gcc-arm-linux-gnueabihf

sudo apt-get install gcc-aarch64-linux-gnu device-tree-compiler lzop libncurses5-dev libssl1.0.0 libssl-dev

# Ubuntu 16.04安装/更新VMware tools&设置共享文件夹方法

2016年12月30日 21:01:12

阅读数：5535

## 环境

虚拟机：VMware Workstation 12 Player，版本12.5.1 build-4542065   
系统：Ubuntu 16.04 LTS

## 安装/更新VMware tools

1、VMware Workstation 12 Player中：Player -> 管理 -> 更新VMware Tools…此时player会打开虚拟CD驱动器（/media目录），里面有VMwareTools的压缩包VMwareTools-xxx-xxx.tar.gz   
2、将VMwareTools-xxx-xxx.tar.gz复制到Home目录下（在光驱中只读），通过终端命令解压并访问其中的内容：

sudo tar zxf VMwareTools-xxx.tar.gz #解压

cd /vmware-tools-distrib #进入文件夹

sudo ./vmware-install.pl #运行安装脚本

* 1
* 2
* 3

它会先卸vmware tools 然后再安装，其中有很多提示，一直按Enter或者输入【】提示yes和no，直到安装完毕。

## 设置共享文件夹

为了实现虚拟机与宿主机的文件交互，可以设置共享文件夹方便地互相访问文件。

【貌似设置共享文件夹必须安装VMware tools】

1、在宿主机创建一个共享文件夹，如D:/share   
2、VMware Workstation 12 Player中：Player -> 管理 -> 虚拟机设置，在“选项”选项卡中设置文件夹共享为D:/share，并总是启用   
3、设置的共享文件夹在Ubuntu中位于/mnt/hgfs下，可通过终端命令进入，或桌面点击：Computer -> mnt -> hgfs查看，成功的话会看到共享文件夹share

## 设置文件夹快捷方式到桌面

普通文件夹可以直接右击 -> make link 创建快捷方式，但貌似computer根目录下的文件夹无法使用，但可以通过命令设置：

ln -s /mnt/hgfs/share /home/wkw1125/share

* 1

以上命令为/mnt/hgfs/share文件夹创建一个名为share快捷方式，位于home文件夹，剪切这个share快捷方式到任意位置（如desktop）即可访问。

**github常见操作和常见错误！错误提示：fatal: remote origin already exists.**

2013年08月29日 15:26:53

阅读数：105479

 原文链接：http://blog.csdn.net/dengjianqiang2011/article/details/9260435

 如果输入$ git remote add origin [git@github.com:djqiang（github帐号名）/gitdemo（项目名）.git](mailto:git@github.com:djqiang/gitdemo.git)

    提示出错信息：fatal: remote origin already exists.

    解决办法如下：

    1、先输入$ git remote rm origin

    2、再输入$ git remote add origin [git@github.com:djqiang/gitdemo.git](mailto:git@github.com:djqiang/gitdemo.git) 就不会报错了！

    3、如果输入$ git remote rm origin 还是报错的话，error: Could not remove config section 'remote.origin'. 我们需要修改gitconfig文件的内容

    4、找到你的github的安装路径，我的是C:\Users\ASUS\AppData\Local\GitHub\PortableGit\_ca477551eeb4aea0e4ae9fcd3358bd96720bb5c8\etc

    5、找到一个名为gitconfig的文件，打开它把里面的[remote "origin"]那一行删掉就好了！

上面无非就是将本地工程同步到github上， 但再使用git remote add origin的时候可能会出现的如下错误

$ git remote addorigin git@github.com:xuxinpie/springDemo.git

错误提示：fatal: remote origin already exists.

解决办法：

$ git remote rm origin

然后在执行：$ git remote add origin git@github.com:xuxinpie/springDemo.git 就不会报错误了

如果输入$ git remote rm origin 还是报错的话，error: Could not remove config section 'remote.origin'. 我们需要修改gitconfig文件的内容

找到你的github的安装路径，我的是C:\Users\ASUS\AppData\Local\GitHub\PortableGit\_ca477551eeb4aea0e4ae9fcd3358bd96720bb5c8\etc  
找到一个名为gitconfig的文件，打开它把里面的[remote "origin"]那一行删掉就好了！

7/问题)编译android时候出错：

**ninja: build stopped: subcommand failed.  
build/core/ninja.mk:142: recipe for target 'ninja\_wrapper' failed  
make: \*\*\* [ninja\_wrapper] Error 1**

**1. 切换到工程根目录，检查 jack-server 是否开启 -------------   ./prebuilts/sdk/tools/jack-admin    list-server  
  
2. 检查宿主目录下的 .jack-server 文件夹以及文件jack-setting  ------------  ls  -al  
如果两文件的所属组与所有者不是编译工程的用户，直接rm 删除  
到prebuilts/sdk/tools 目录下执行一下这个命令：./jack-admin install-server jack-launcher.jar jack-server-4.8.ALPHA.jar  
jack-launcher.jar，jack-server-4.8.ALPHA.jar这两个文件的名字视 prebuilts/sdk/tools 目录下的文件名而定。  
  
3. 执行完毕后开启服务  prebuilts/sdk/tools/jack-admin  start -server，执行成功就ok  
Launching Jack server java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=/tmp -Dfile.encoding=UTF-8 -XX:+TieredCompilation -cp /home/xxx(用户名)/.jack-server/launcher.jar com.android.jack.launcher.ServerLauncher  
  
4. 接着编译 make -j8 ,线程加多少个具体看机器配置，问题也最可能是这一步骤引起的，如果是虚拟机的话，建议不要加线程，  
直接make 走起，跟楼主遇见的问题一模一样，后来人别再纠结了**

**解决方法网址：**

**https://bbs.csdn.net/topics/392023688**

[ 1% 540/45962] host C++: libchrome\_32 <= external/libchrome/base/vlog.ccninja: fatal: fork: Cannot allocate memory

build/core/ninja.mk:148: recipe for target 'ninja\_wrapper' failed

make: \*\*\* [ninja\_wrapper] Error 1

解决方法：

Q5.报错fatal: fork: Cannot allocate memory (Ubuntu17.04-Android7.1)

关键Log:

|  |
| --- |
| [  3% 802/22494] Install: out/target/p...duct/angler/system/app/Music/Music.apk  ninja: fatal: fork: Cannot allocate memory  build/core/ninja.mk:148: recipe for target ‘ninja\_wrapper‘ failed  make: \*\*\* [ninja\_wrapper] Error 1 |

解决方案：

|  |
| --- |
| **无法分配内存(2次)** |
| **方法1** |
| **sysctl kernel.pid\_max   // 查看最大进程数**  **ps -eLf | wc –l   // 查看运行进程数**  **// 修改最大进程数后系统恢复**  **echo 1000000 > /proc/sys/kernel/pid\_max**  **// 永久生效**  **echo "kernel.pid\_max=1000000 " >> /etc/sysctl.conf**  **sysctl –p**  **结果：无权限修改最大进程数** |
| **方法2** |
| **查看虚拟机内存分配只有6G左右，关闭虚拟机内存调整到8G，编译正常。** |

Ref:

[-bash: fork: Cannot allocate memory 问题的处理](http://blog.csdn.net/woshizhangliang999/article/details/66473884)

解决网址：

<http://www.bubuko.com/infodetail-2118922.html>

**编译出错 ninja: build stopped: subcommand failed Android 源码编译**

2018年06月05日 14:48:02

阅读数：108

标签：ulimit -c unlimited 、core文件、Jack、JACK\_SERVER\_VM\_ARGUMENTS  
  
错误：ninja: build stopped: subcommand failed.  
办法一：按照提示：ulimit -c unlimited  
该命令的详细解释参考：https://www.cnblogs.com/qq78292959/archive/2012/05/08/2490443.html  
  
命令解释：  
ulimint -a 用来显示当前的各种用户进程限制  
Linux对于每个用户，系统限制其最大进程数，为提高性能，可以根据设备资源情况，  
设置个Linux用户的最大进程数，一些需要设置为无限制：  
数据段长度无限制：ulimit -d unlimited  
最大内存大小无限制：ulimit -m unlimited  
堆栈大小无限制：ulimit -s unlimited  
core文件大小无限制：ulimit -c unlimited   
  
core文件的简单介绍 ：  
在一个程序崩溃时，它一般会在指定目录下生成一个core文件。core文件仅仅是一个内存映象(同时加上调试信息)，主要是用来调试的。  
当系统中的一些程序在遇到一些错误以及crash时，系统会自动产生core文件记录crash时刻系统信息，包括内存和寄存器信息，用以程序员日 后debug时可以使用。这些错误包括段错误、非法指令、总线错误或用户自己生成的退出信息等等，一般地，core文件在当前文件夹中存放。  
core文件有时可能在你发生错误时，并没有出现在你当前的文件夹中，发生这种情况的原因有两个：一个是当前终端被设置为不能弹出core文件；另一种则是core文件被指定了路径。  
对于前者，我们可以使用ulimit命令对core文件的大小进行设定。一般默认情况下，core文件的大小被设置为0，这样系统就不dump出core文件了。这时，使用命令：ulimit -c unlimited进行设置，就可以把core文件的大小设置为无限大，同时也可以使用数字来替代unlimited，对core文件的上限制做更精确的设定。  
  
  
办法二，修改环境变量：  
参考：https://blog.csdn.net/bobcat\_kay/article/details/76564969?locationNum=10&fps=1，键入命令：  
export JACK\_SERVER\_VM\_ARGUMENTS="-Dfile.encoding=UTF-8 -XX:+TieredCompilation -Xmx4096m"  
out/host/linux-x86/bin/jack-admin kill-server  
out/host/linux-x86/bin/jack-admin start-server  
  
办法三，修改Jack配置文件：  
背景知识：什么是Jack  
参考：https://blog.csdn.net/xz10561/article/details/53886691  
简单了解：  
我们先来看一下关于Jack server官方介绍：  
Jack is a new Android toolchain that compiles Java source into Android dex bytecode. It replaces the previous Android toolchain, which consists of multiple tools, such as javac, ProGuard, jarjar, and dx.   
（Jack编译器工具是一套新的Android编译工具用来将Java代码转换为Android dex字节代码。）  
  
参考：https://blog.csdn.net/qq\_27061049/article/details/70156200  
已经做出的修改：  
1）、修改prebuilts/sdk/tools/jack-admin 文件，变量JACK\_SERVER\_VM\_ARGUMENTS，添加参数 -Xmx4096M  
JACK\_SERVER\_VM\_ARGUMENTS  
=”${JACK\_SERVER\_VM\_ARGUMENTS:=-Dfile.encoding=UTF-8 -XX:+TieredCompilation -Xmx4096}”  
  
2）、prebuilts/sdk/tools/jack-admin 文件，在 JACK\_SERVER\_COMMAND=“…… -cp ……” -cp 前添加-Xmx4096m（4096m为内存大小的一半左右，根据所用的主机配置而定）  
  
JACK\_SERVER\_COMMAND="java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=$TMPDIR $JACK\_SERVER\_VM\_ARGUMENTS -Xmx4096m -cp $LAUNCHER\_JAR $LAUNCHER\_NAME"  
         
3）、prebuilts/sdk/tools/README-jack-server.md  
-JACK\_SERVER\_VM\_ARGUMENTS="-Xmx2g -Dfile.encoding=UTF-8 -XX:+TieredCompilation" jack-admin start-server  
+JACK\_SERVER\_VM\_ARGUMENTS="-Xmx4g -Dfile.encoding=UTF-8 -XX:+TieredCompilation" jack-admin start-server

[ 13% 4509/33901] Building with Jack: ...k\_intermediates/with-local/classes.dex

FAILED: /bin/bash out/target/common/obj/JAVA\_LIBRARIES/framework\_intermediates/with-local/classes.dex.rsp

Out of memory error (version 1.2-rc4 'Carnac' (298900 f95d7bdecfceb327f9d201a1348397ed8a843843 by android-jack-team@google.com)).

GC overhead limit exceeded.

Try increasing heap size with java option '-Xmx<size>'.

Warning: This may have produced partial or corrupted output.

[ 13% 4509/33901] Building with Jack: ...RIES/volley\_intermediates/classes.jack

ninja: build stopped: subcommand failed.

build/core/ninja.mk:148: recipe for target 'ninja\_wrapper' failed

make: \*\*\* [ninja\_wrapper] Error 1

# 编译Android7源码出现:ninja: build stopped: subcommand failed.错误

2018年06月12日 10:45:45

阅读数：25

**[plain]** [view plain](https://blog.csdn.net/zhaohuarong/article/details/80662197) [copy](https://blog.csdn.net/zhaohuarong/article/details/80662197)

1. $ vi prebuilts/sdk/tools/jack-admin

1.找到如下语句：  
JACK\_SERVER\_COMMAND="java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=$TMPDIR $JACK\_SERVER\_VM\_ARGUMENTS -cp $LAUNCHER\_JAR $LAUNCHER\_NAME"  
2.将上述语句修改为：  
JACK\_SERVER\_COMMAND="java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=$TMPDIR $JACK\_SERVER\_VM\_ARGUMENTS -Xmx4096m -cp $LAUNCHER\_JAR $LAUNCHER\_NAME"

主要是添加了-Xmx4096m参数，接下来在源码目录下执行如下命令重启jack-admin服务：

**[plain]** [view plain](https://blog.csdn.net/zhaohuarong/article/details/80662197) [copy](https://blog.csdn.net/zhaohuarong/article/details/80662197)

1. $ ./prebuilts/sdk/tools/jack-admin stop-server
2. $ ./prebuilts/sdk/tools/jack-admin start-server

再重新make就可以了。

解决问题的网址：

https://blog.csdn.net/zhaohuarong/article/details/80662197

ninja: build stopped: subcommand failed.

build/core/ninja.mk:148: recipe for target 'ninja\_wrapper' failed

make: \*\*\* [ninja\_wrapper] Error 1

java.io.IOException: No space left on device

at java.io.FileOutputStream.writeBytes(Native Method)

at java.io.FileOutputStream.write(FileOutputStream.java:326)

at com.android.signapk.SignApk.main(SignApk.java:1154)

ninja: build stopped: subcommand failed.

build/core/ninja.mk:148: recipe for target 'ninja\_wrapper' failed

make: \*\*\* [ninja\_wrapper] Error 1

**编译安卓7.0源码出现make: \*\*\* [ninja\_wrapper] Error 1**

2017年07月28日 15:41:24

阅读数：4478

32 warnings generated.  
ninja: build stopped: subcommand failed.  
build/core/ninja.mk:148: recipe for target 'ninja\_wrapper' failed  
make: \*\*\* [ninja\_wrapper] Error 1

1，在jack-admin 中加入了 -Xmx8192m

修改prebuilts/sdk/tools/jack-admin 文件，变量JACK\_SERVER\_VM\_ARGUMENTS，在 -cp 前添加参数 -Xmx8192m

然后重新启动Jack server.

@flm:~/android1/alps/prebuilts/sdk/tools$ ./jack-admin stop-server  
Stopping background server  
@flm:~/android1/alps/prebuilts/sdk/tools$ 

@flm:~/android1/alps/prebuilts/sdk/tools$ ./jack-admin start-server

Launching Jack server java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=/tmp -Dfile.encoding=UTF-8 -XX:+TieredCompilation -Xmx8192m -cp /home/archermind/.jack-server/launcher.jar com.android.jack.launcher.ServerLauncher  
@flm:~/android1/alps/prebuilts/sdk/tools$

# AndroidN 编译代码 make: \*\*\* [ninja\_wrapper] Error 1

2018年04月09日 12:01:35

阅读数：45

最近在编译RK3399J的AndroidN的源码，编译的时候遇到 make: \*\*\* [ninja\_wrapper] Error 1的问题，网上搜索了一下，找到这个 [点击打开链接](https://stackoverflow.com/questions/34940793/increasing-heap-size-while-building-the-android-source-code-on-ubuntu-15-10)

export JACK\_SERVER\_VM\_ARGUMENTS="-Dfile.encoding=UTF-8 -XX:+TieredCompilation -Xmx4096m"

out/host/linux-x86/bin/jack-admin kill-server

out/host/linux-x86/bin/jack-admin start-server

# 系统编译错误记录

2016年06月29日 11:10:44

阅读数：1325

//Android.mk文件中添加LOCAL\_STATIC\_JAVA\_LIBRARIES := android-support-v4

//系统app不需要versionCode ,versionCode ,minSdkVersion ,targetSdkVersion  删除即可

Warning: AndroidManifest.xml already defines versionCode (in http://schemas.android.com/apk/res/android); using existing value in manifest.  
Warning: AndroidManifest.xml already defines versionCode (in http://schemas.android.com/apk/res/android); using existing value in manifest.  
Warning: AndroidManifest.xml already defines minSdkVersion (in http://schemas.android.com/apk/res/android); using existing value in manifest.  
Warning: AndroidManifest.xml already defines targetSdkVersion (in http://schemas.android.com/apk/res/android); using existing value in manifest.  
  
  
//系统编译xml文件时候，textView里面的text属性必须用@string引用不能直接写字符串  
/home/myzr/myandroid/packages/apps/AplexTest/res/layout/ui\_buzz.xml:43: error: Error: This attribute must be localized. (at 'text' with value '500').  
/home/myzr/myandroid/packages/apps/AplexTest/res/layout/ui\_buzz.xml:58: error: Error: This attribute must be localized. (at 'text' with value '1000').  
/home/myzr/myandroid/packages/apps/AplexTest/res/layout/ui\_buzz.xml:73: error: Error: This attribute must be localized. (at 'text' with value '1500').

/home/myzr/myandroid/packages/apps/AplexTest/res/layout/ui\_buzz.xml:88: error: Error: This attribute must be localized. (at 'text' with value '2000').

//编译出来没有.dex文件   解决：在Android.mk文件中添加jar支持包的路径 LOCAL\_STATIC\_JAVA\_LIBRARIES := android-support-v4

**[html]** [view plain](https://blog.csdn.net/u011168565/article/details/51781571) [copy](https://blog.csdn.net/u011168565/article/details/51781571)

1. LOCAL\_PATH:= $(call my-dir)
2. include $(CLEAR\_VARS)
4. LOCAL\_MODULE\_TAGS := optional
6. LOCAL\_STATIC\_JAVA\_LIBRARIES := android-support-v4

**[html]** [view plain](https://blog.csdn.net/u011168565/article/details/51781571) [copy](https://blog.csdn.net/u011168565/article/details/51781571)

1. //从src目录下开始编译 切记

**[html]** [view plain](https://blog.csdn.net/u011168565/article/details/51781571) [copy](https://blog.csdn.net/u011168565/article/details/51781571)

1. LOCAL\_SRC\_FILES := $(call all-java-files-under, src)
2. //LOCAL\_SRC\_FILES := $(call all-java-files-under, com)

5. LOCAL\_PACKAGE\_NAME := AplexTest
6. LOCAL\_CERTIFICATE := platform
8. include $(BUILD\_PACKAGE)

11. # Use the folloing include to make our test apk.
12. include $(call all-makefiles-under,$(LOCAL\_PATH))

# jack server 常见错误解决方法

2017年04月13日 11:14:05

阅读数：13671

## jack 服务常见错误解决方法

　　当你编译Android时，你不需要修改任何内容。 Jack是Andriod M的默认编译工具。只需使用标准的makefile命令执行即可。当第一次执行jack时，它会在你的机器上启动一个jack编译服务：   
　　这个服务大大提高了编译速度，因为它避免了启动一个新的JVM，加载Jack代码，初始化Jack和每次编译都要预制JIT，即使在很小的编译过程中，它也很好的节省了时间（例如，在incremental模式）   
这个服务也是一个控制Jack并行编译的短期解决方案。它可以限制编译的并行数据量，这样也可以避免内存硬盘溢出问题   
当一段时间没有编译活动，Jack 服务会自动关闭，默认它在本地使用了两个TCP端口，且不对外公开。它也提供了很多参数来配置Jack服务，例如并行数量，timeout时间，端口等等。

### 1、错误显示

　　Android 7.0工程的环境需要加载openJDK 8 ，虽然上篇文档已经描述了openjdk-8的安装方法，但通常和很多ubuntu主机的Jack -server是没有和Android7.0的工程Jack -server相互交互的，Android的工程会提供相应的压缩文件去运行jack-server服务，若要顺利的编译工程,则要ubuntu主机的Jack -server和Android的工程Jack -server相互交互。

在编译过程中会有如下错误信息。   
[6% 1127/18030] Ensure Jack server is installed and started

FAILED: /bin/bash -c “(prebuilts/sdk/tools/jack-admin install-server prebuilts/sdk/tools/jack-launcher.jar prebuilts/sdk/tools/jack-server-4.8.ALPHA.jar 2>&1 || (exit 0) ) && (JACK\_SERVER\_VM\_ARGUMENTS=/”-Dfile.encoding=UTF-8

No Jack server running. Try ‘jack-admin start-server’   
No Jack server running. Try ‘jack-admin start-server’

bule@sky:~/\*\*/jianwen.fu/V65\_An7/prebuilts/sdk/tools$ jack-admin start-server   
Launching Jack server java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=/tmp -Dfile.encoding=UTF-8 -XX:+TieredCompilation -cp /home/prayasm/.jack-server/launcher.jar com.android.jack.launcher.ServerLauncher   
Jack server failed to (re)start, try ‘jack-diagnose’ or see Jack server log

### 2、ubuntu主机jack相关配置

.ubuntu主机的本地目录的

　　注意在服务器192.168.2.188主目录不存在jack-server，我以自己的主机模拟192.168.2.188环境，下面的命令作为参考；

找个已安装openjdk8环境的Ubuntu主机，把文件拷贝过来，这样的方法比以前编译openjdk8的方法更快速，但依赖关系并不好，风险度高。（也可以自己安装，安装链接<http://blog.csdn.net/qq_27061049/article/details/65626930>）

1）、 scp -r /etc/java-8-openjdk/ bule@192.168.2.244:/etc/   
（先传输该文件夹，下面的文件夹中有对此文件的软连接文件）   
2）、 scp -r /usr/lib/jvm/java-1.8.0-openjdk-amd64 bule@192.168.2.244:/usr/lib/jvm   
3）、scp -r /usr/lib/jvm/java-8-openjdk-amd64/ bule@192.168.2.244:/usr/lib/jvm

如服务器192.168.2.188，Ubuntu用户主目录无.jack .jack-server .jack-setting，那么在192.168.1.15的Ubuntu主机中执行以下命令，边将文件传输到了192.168.2.244主机上。

4）、 scp -r .jack-server/ bule@192.168.2.244:.jack-server   
5）、scp -r .jack bule@192.168.2.244:.jack （这个文件很多主机没有）

6）、scp -r .jack-settings bule@192.168.2.244:.jack-settings

（ubuntu 主机 jack 服务相关文件下载地址：<http://download.csdn.net/detail/qq_27061049/9817787>）

将openjdk写入系统的环境和配置编译选项中

7)、 sudo update-alternatives –install /usr/bin/java java /usr/lib/jvm/java-8-openjdk-amd64/bin/java 1081   
8)、 sudo update-alternatives –install /usr/bin/java java /usr/lib/jvm/java-8-openjdk-amd64/bin/java 1081   
9)、 sudo update-alternatives –install /usr/bin/javac javac /usr/lib/jvm/java-8-openjdk-amd64/bin/javac 1081   
10)、sudo update-alternatives –install /usr/bin/javap javap /usr/lib/jvm/java-8-openjdk-amd64/bin/javap 1081   
11)、 sudo update-alternatives –install /usr/bin/javah javah /usr/lib/jvm/java-8-openjdk-amd64/bin/javah 1081   
12)、 sudo update-alternatives –install /usr/bin/jar jar /usr/lib/jvm/java-8-openjdk-amd64/bin/jar 1081   
13)、 sudo update-alternatives –install /usr/bin/javadoc javadoc /usr/lib/jvm/java-8-openjdk-amd64/bin/javadoc 1081

转换为openjdk-8-jre的java环境

14）、 sudo update-alternatives –config javadoc   
15）、sudo update-alternatives –config java   
16）、 sudo update-alternatives –config javac   
17） sudo update-alternatives –config javap   
18）、 sudo update-alternatives –config javah   
19）、 sudo update-alternatives –config javadoc   
20）、 sudo update-alternatives –config jar

### 3、源码工程jack相关配置

在Android 7.0的工程中进行编译之前，运行如下命令，进行交互。   
进入源码相应目录 prebuilts/sdk/tools$   
运行下面命令：   
1）、 jack-admin start-server   
2）、jack-admin kill-server   
3）、jack-admin list-server   
4）、jack-admin uninstall-server   
5）、mm -j32 showcommands &> mm.out   
6）、jack-admin install-server jack-launcher.jar jack-server-4.8.ALPHA.jar   
7）、jack-admin dump-report   
8）、jack-admin dump-re

### 4、若没有解决尝试其他相关修改：

1）、修改prebuilts/sdk/tools/jack-admin 文件，变量JACK\_SERVER\_VM\_ARGUMENTS，添加参数 -Xmx4096M

JACK\_SERVER\_VM\_ARGUMENTS=”${JACK\_SERVER\_VM\_ARGUMENTS:=-Dfile.encoding=UTF-8 -XX:+TieredCompilation -mx4096M}”

2）、prebuilts/sdk/tools/jack-admin 文件，在 JACK\_SERVER\_COMMAND=“…… -cp ……” -cp 前添加-Xmx4096m（4096m为内存大小的一半左右，根据所用的主机配置而定）

diff --git a/tools/jack-admin b/tools/jack-admin

index ee193fc..33cb913 100755

--- a/tools/jack-admin

+++ b/tools/jack-admin

@@ -451,7 +451,7 @@ case $COMMAND in

if [ "$RUNNING" = 0 ]; then

echo "Server is already running"

else

- JACK\_SERVER\_COMMAND="java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=$TMPDIR $JACK\_SERVER\_VM\_ARGUMENTS -cp $LAUNCHER\_JAR $LAUNCHER\_NAME"

+ JACK\_SERVER\_COMMAND="java -XX:MaxJavaStackTraceDepth=-1 -Djava.io.tmpdir=$TMPDIR $JACK\_SERVER\_VM\_ARGUMENTS -Xmx4096M -cp $LAUNCHER\_JAR $LAUNCHER\_NAME"

echo "Launching Jack server" $JACK\_SERVER\_COMMAND

(

trap "" SIGHUP

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13

3）、prebuilts/sdk/tools/README-jack-server.md

* -JACK\_SERVER\_VM\_ARGUMENTS="-Xmx2g -Dfile.encoding=UTF-8 -XX:+TieredCompilation" jack-admin start-server   
  +JACK\_SERVER\_VM\_ARGUMENTS="-Xmx4g -Dfile.encoding=UTF-8 -XX:+TieredCompilation" jack-admin start-server

4）、build/core/configure\_local\_jack.mk   
by changing the line:   
LOCAL\_JACK\_ENABLED := (strip(strip(LOCAL\_JACK\_ENABLED))   
into:   
LOCAL\_JACK\_ENABLED := #(strip(strip(LOCAL\_JACK\_ENABLED))   
““““““““““““““““““““““““““““““““““““““`   
5）、把文件$HOME/.jack-server/config.properties的权限由rwx改为rw

6）、同时修改HOME/.jack−settings和HOME/.jack−settings和HOME/.jack-server/config.properties中的端口号（比如都改为8086/8087），方可支持多用户同时编译。

7）、当在板型目录中的BoardConfig.mk中添加ANDROID\_COMPILE\_WITH\_JACK:=false / DEFAULT\_JACK\_ENABLED := disabled / ANDROID\_FORCE\_JACK\_ENABLED := disabled来进行关闭JACK   
8） 或者在运行编译命令时加入 make -j8 ANDROID\_COMPILE\_WITH\_JACK=false 来关闭jack 服务   
9)、添加相应jack-admine 的路径 ，在相应的代码目录下 PATH=./prebuilts/sdk/tools/:$PATH   
10）、可以通过运行jack-admin server-log命令来查找jack log 具体分析

文档参考：   
Android 7.0工程之预编译jack服务器   
<http://www.th7.cn/system/lin/201702/202334.shtml>

Android7.0 JACK编译器不支持多用户同时编译的问题的解决   
<http://blog.csdn.net/lizekun2010/article/details/52535591>

使用Jack编译   
<http://blog.csdn.net/pcsxk/article/details/52213784#t5>

android N编译   
<http://blog.csdn.net/luvzhan/article/details/53282968>

**vmware ubuntu硬盘空间不够用，空间扩展**

2016年12月25日 11:16:18

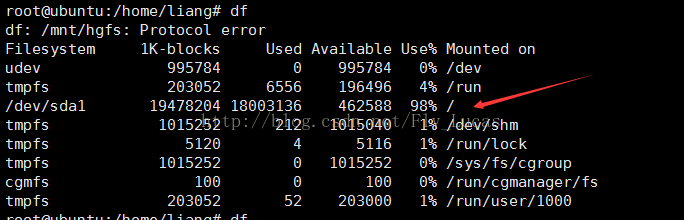
阅读数：9432

我从来没有想过我的虚拟机内存会不够用，毕竟已经20G了，可是最近学习python，装了些学习有关的软件，

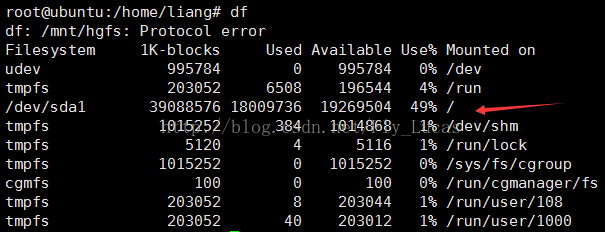
期末做libvirt管理实验，存了两个镜像，就变成这样了，所以，我就像了要扩展硬盘空间，在网上看到了这个

Gparted工具。就尝试了下，最后成功了。。。

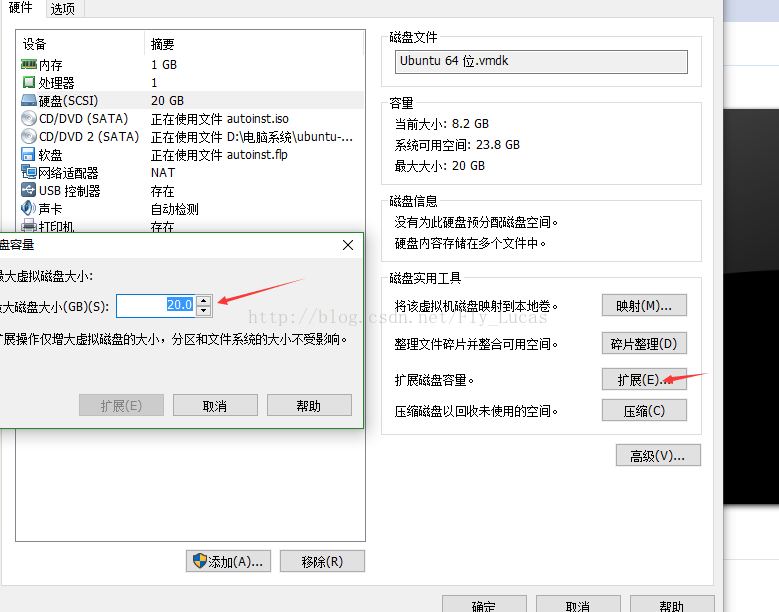
未扩展前：



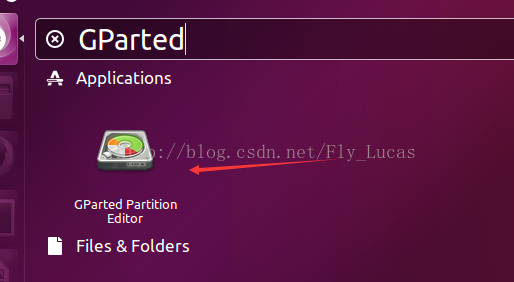
扩展后：



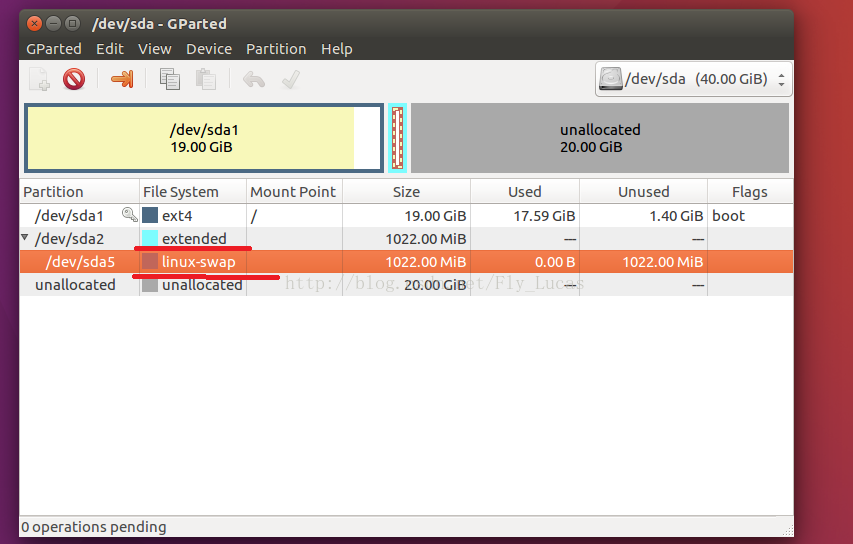
1.在vmware中选择虚拟机设置，输入你想扩展到的大小，最后点击扩展。



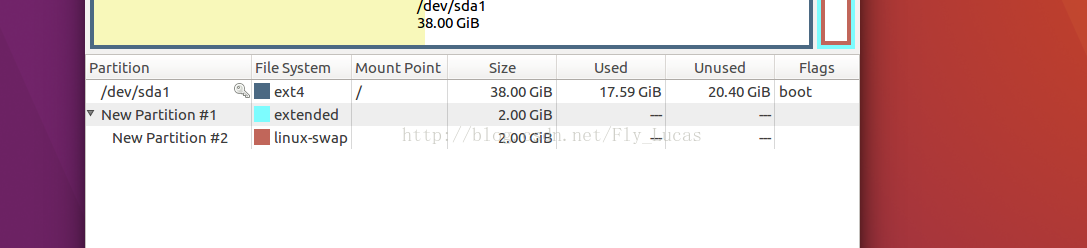
1. 安装gparted分区管理软件，apt-get install gparted
2. 下载好之后重启虚拟机



3.打开软件，将extend分区删掉，不删的话无法重置sda1主分区大小



4.重置sda1的大小后，再次将新建一个一个extend分区,然后在extend分区下在建一个swap分区，点击上方绿色的对号保存即可



扩展Linux分区网址：

https://blog.csdn.net/fly\_lucas/article/details/53868722

**Android cannot allocate memory**

2017年03月31日 17:54:44

阅读数：805

之前git上切分支编译android代码时，虚拟机总是出现cannot allocate memory问题，后来整理了下解决方法：

1.创建swapfile文件  
下面的代码是创建一个1G的swapfile，如果想要512M，将bs将为1024  
sudo dd if=/dev/zero of=/swapfile bs=2048 count=524288  
ls -lh /swapfile   //查看是否创建成功  
  
2.设置交换区  
sudo mkswap /swapfile  
  
3.启动系统交换区  
sudo swapon /swapfile

如果以上方法不能解决问题，那就只有make clean，全编了。Android系统代码复杂，建议大家好好管理代码。

另外扩展交换内存可以直接使用Linux gparted软件进行扩展！简单方便

**-bash: fork: Cannot allocate memory 问题的处理**

2017年03月26日 12:58:47

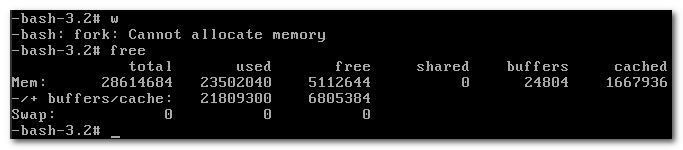
阅读数：20692

文章来源：**[fork:cannot allocate memory问题的处理](http://bbs.qcloud.com/thread-22637-1-1.html" \t "_blank)**

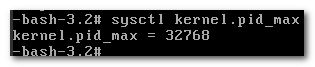
文章参考：[**pid max导致fork: Cannot allocate memory 的分析及解决办法**](http://blog.csdn.net/onlyellow/article/details/51917757)

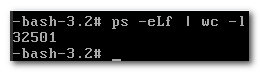
今天遇到服务器无法SSH，VNC操作命令提示fork:cannot allocate memory

free查看内存还有（注意，命令可能要多敲几次才会出来）



查看最大进程数 sysctl kernel.pid\_max



ps -eLf | wc -l查看进程数  
   
  
确认是进程数满了  
  
修改最大进程数后系统恢复  
echo 1000000 > /proc/sys/kernel/pid\_max  
  
永久生效  
echo "kernel.pid\_max=1000000 " >> /etc/sysctl.conf  
sysctl -p