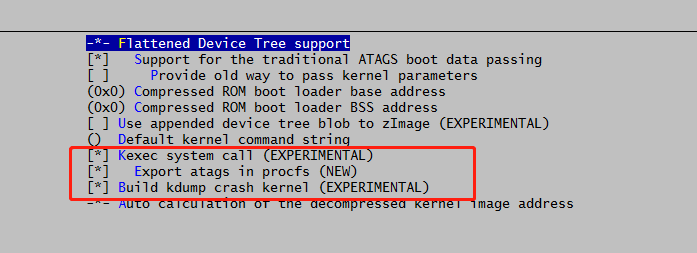
参考文章：

<https://biscuitos.github.io/blog/CRASH/>

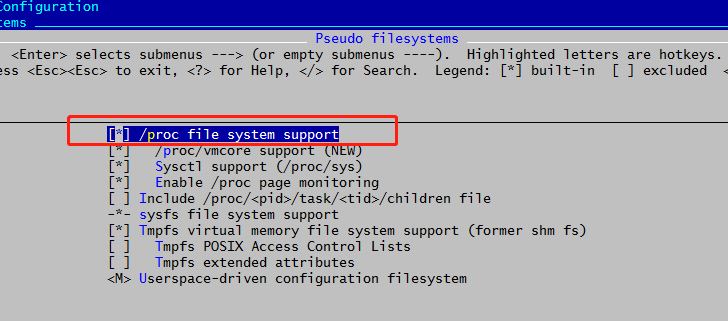
目前主流的方法获得 core dump 是通过 kexec 工具，该工具当内核核心转储时，其他 CPU 夯住，kexec 工具会在一个预留 CPU 上重启一个精简的内核。当精简内核启动之后，可以通过一定的命令将 core 写入到指定的文件，该文件就是核心转储文件。当获得核心转储文件之后，可以使用 CRASH 等工具进行问题分析。

Makemenuconfig:

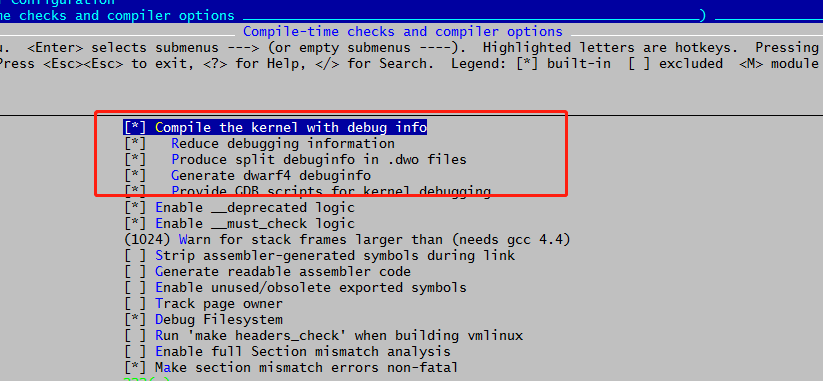
**CONFIG\_KEXEC:**



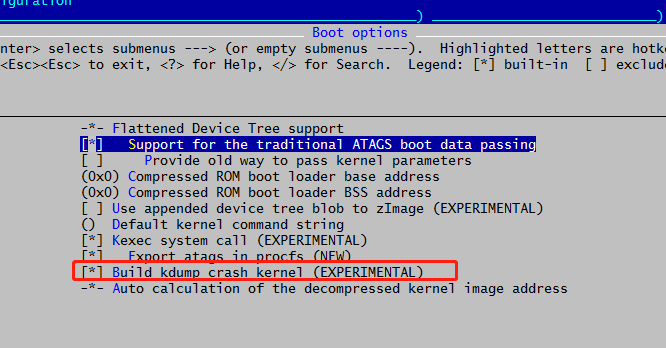
**CONFIG\_SYSFS:**



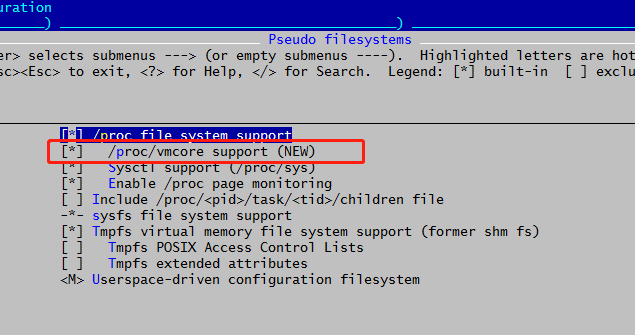
**CONFIG\_DEBUG\_INFO:**



**CONFIG\_CRASH\_DUMP:**



**CONFIG\_PROC\_VMCORE:**



**内核转储 Kdump CMDLINE 配置：**

内核启用相关的内核配置之后，重新编译内核之后需要在 CMDLINE 中配置 crash 内核占用的内存大小，其语法如下:

setenv bootargs\_sys1 'console=ttyS0,115200 mem=512M root=/dev/mmcblk1p1 rw rootwait init=/sbin/init earlyprintk crashkernel=128M'

**内核转储工具安装使用：**

git clone <http://kernel.org/pub/linux/utils/kernel/kexec/kexec-tools.tar.gz>

tar xf kexec-tools.tar.gz

cd kexec-tools-VERSION

./configure ARCH**=**arm --build**=**x86\_64-unknown-linux-gnu --host**=**arm-linux-gnueabi LDFLAGS**=**-static

make

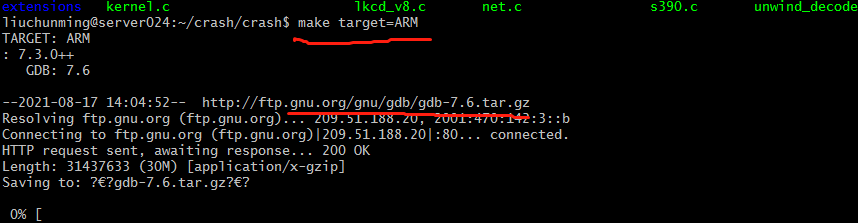
将build目录下的sbin及man copy到开发板的环境变量目录中。

**Crash**

git clone https://gitee.com/mirrors/crash.git

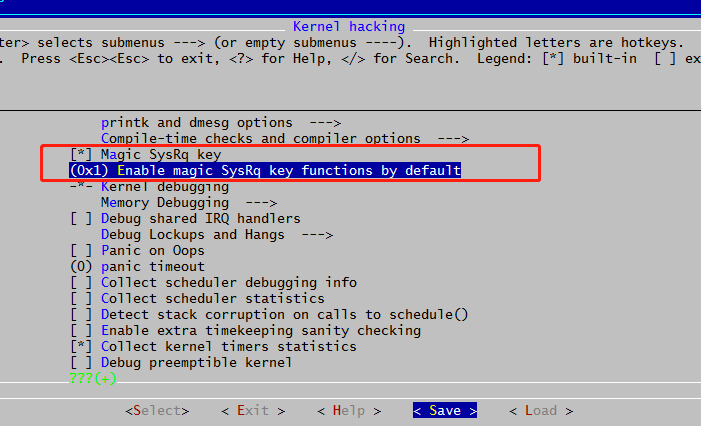
git clone <https://github.com/crash-utility/crash.gi>

make target**=**ARM



./configure ARCH**=**arm --build**=**x86\_64-linux-gnu --host**=** arm-xilinx-linux-gnueabi LDFLAGS**=**-static

出发异常用：



kexec -p <Dump-capture-kernel> --initrd=<initrd-for-Dump-capture-kernel> \

--append=<CMDLINE>



kexec --type zImage -p /data/zImage --dtb=/data/devicetree.dtb --append="ver=v001 console=ttyS0,115200 mem=256M root=/dev/mmcblk1p1 rw rootwait init=/sbin/init earlyprintk product=npes01h\_pro\_ant4\_S"