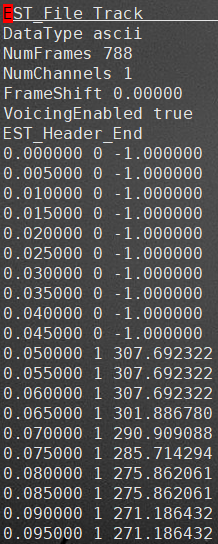
# 提取参数方法

运行get\_f0.sh使用reaper工具获取f0基频，reaper工具输出为

NumFrames为共有多少帧，后面的三数字组成的一组数据，第二个是是否为静音，第三个为基频f0.，get\_f0.sh提取这三个数字对应每帧的三个数据



运行f0\_stats\_pic.py输出每个wav文件的最小基频、最大基频、平均基频、标准差。最终输出最大基频和最小基频

为得到准确f0，再次运行get\_f0.sh文件

30~800:33~728

执行MakeFile文件，make sampwav是将wav中的wav文件用sox工具以16k复制到wav-samp文件夹中

可以直接运行 make all

运行后获得lf0为一维

参数：

FRAMESHIFT——帧移

FRAMELEN——帧长

工具介绍：

SOX——音频格式转化工具<https://blog.csdn.net/y_xianjun/article/details/73245482>

X2X——音频合并工具

PITCH

加上UV信息部分：out\_dimension\_dict用于获取流中不同特征开始位置

UV信息是指vuv

在合成时设置维度为

Reaper+straight:

lf0:1

vuv:1

lsp(mgc):41

# label提取方法

首先对记录label的txt文件去掉韵律标记

del\_prosody\_tag.py

之后使用iconv转码

iconv -f gbk -t utf-8 HW-100001-105000.txt > HW-100001-105000\_utf-8.txt

在线上系统docker中，使用extract\_label提取label，生成文本格式和onehot格式两种label，注意一定要在docker系统中使用

发生Segmentation fault(core dumped)，注意修改代码中的