



Zy & Linux

BUPT 张阳

- 首先说明一下，我是一个微软的粉丝。
- 其实对于Linux，我的了解很有限。我是在大二因为好奇才开始接触Linux的。
- 如今的我依然是个Linux小白，但之前几位学姐问我怎么在服务器上用anaconda
- 因此借助这个机会分享一下我这快一年来使用Linux的经历，希望能给大家带来一些帮助
- 如果有什么不对的话，一定要批评指正



Microsoft

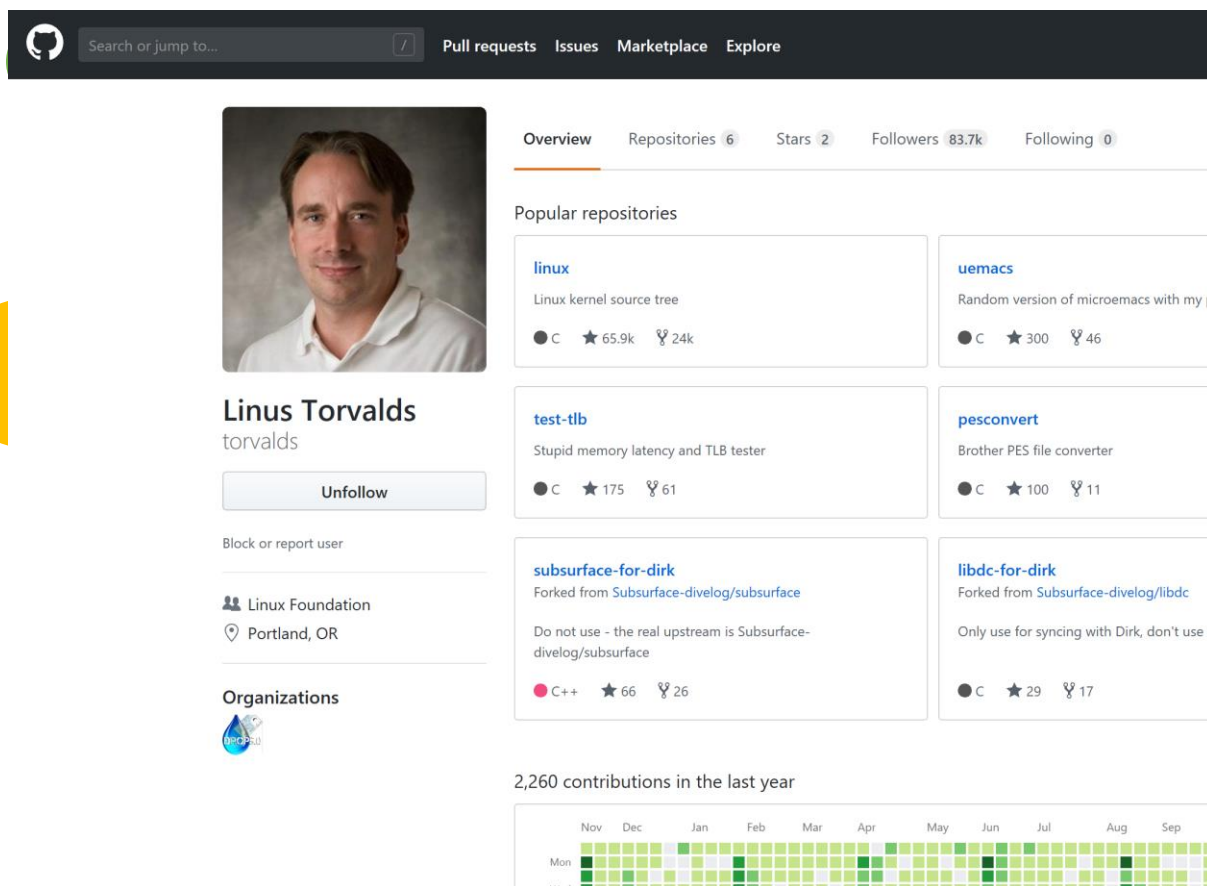
Open source, from Microsoft with love

Redmond, WA <https://opensource.micro...>



Linux的发展

- 林纳斯·托瓦兹 My name is Linus, and I am your God.



The screenshot shows the GitHub profile of Linus Torvalds. At the top, there is a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. The profile header includes a profile picture, the name "Linus Torvalds", and the handle "torvalds". Below this is an "Unfollow" button and a link to "Block or report user". The "Organizations" section shows the Linux Foundation and Portland, OR. The "Overview" tab is selected, displaying statistics: 6 Repositories, 2 Stars, 83.7k Followers, and 0 Following. The "Popular repositories" section lists several projects: "linux" (65.9k stars, 24k forks), "uemacs" (300 stars, 46 forks), "test-tlb" (175 stars, 61 forks), "pesconvert" (100 stars, 11 forks), "subsurface-for-dirk" (66 stars, 26 forks), and "libdc-for-dirk" (29 stars, 17 forks). At the bottom, a heatmap shows "2,260 contributions in the last year" with a grid of green squares representing activity from November to November.



Linux



Linux的发展

- 由于我不是学计算机专业的，这里怕说错，所以写的不是很详细~

● Unix & Linux

- 操作系统有两大阵营，一边是基于微软 Windows NT 的操作系统，一边是由UNIX衍生下来的操作系统。
- Linux, Mac OS X, Android, iOS, Chrome OS 甚至路由器上的固件，这些操作系统同出一族，都是基于最初的UNIX系统开发而来，统称Unix-like 操作系统。

● GNU/Linux

- 其实Linux并不是操作系统。它仅仅是所用系统的一部分。
- Linux是内核：它是为你运行的其他程序分配计算机资源的程序。
- 内核是操作系统的基本部分，但是它自己并无用处；它只能在完整的操作系统框架下才能发挥作用。Linux一般和GNU操作系统一起使用：整个系统基本上就是GNU加上Linux，或叫GNU/Linux。
- （这里还是让专业的蓝天学长解释解释）
- 尽管Linus是神一样的存在，操作系统的开发一个人的力量太过渺小，需要集体的力量
- 所有被叫做“Linux”的发行版实际上是GNU/Linux发行版。



Linux的应用

● Linux的应用领域

1. 企业级IT服务器

例如实验室的服务器的操作系统、Linux还可以方便的搭建分布式系统or集群。Mosos (我认为这也是一个很有趣的领域)

2. 嵌入式Linux系统

例如树莓派这样的arm架构的微机、手机用的安卓也是基于Linux开发的

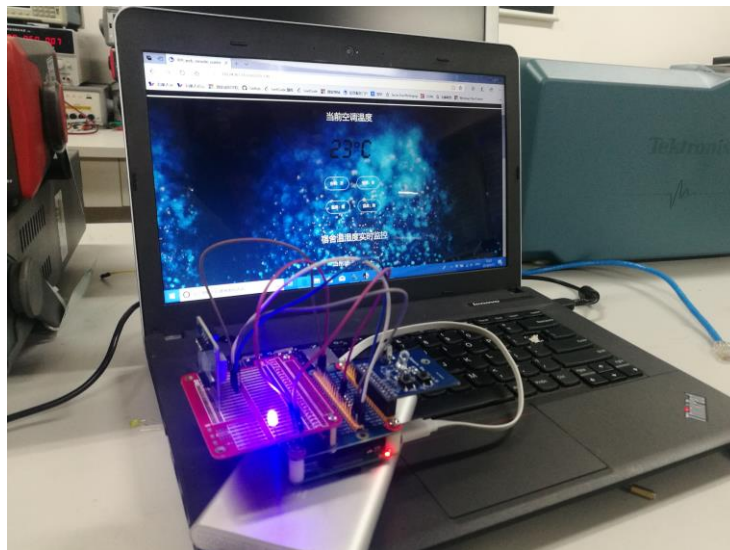
3. 个人桌面Linux应用领域

这个领域还是大微软强，德国政府用了10年的Linux，最后还是换回了windows。

(我个人也认为桌面Linux不好用)
最新的ubuntu18.04LTS好像已经放弃了unity，之前用过感觉一堆bug

● 常见的GNU/Linux发行版

- Centos
- Ubuntu
- Debian
- Arch Linux



Linux的设计哲学

1. 一切皆文件
2. 由众多单一目的的小程序，一个程序只实现一个功能，多个程序组合完成复杂任务
3. 文本文件保存配置信息
4. 尽量避免与用户交互

Linux认为：用户知道自己想要什么,也明白自己在做什么,并且会为自己的行为负责。

而windows恰好相反。

Linux is free only if your time is free

Shell

● 常用的解释器

● Unix/Linux上常见的Shell脚本解释器有bash、sh、csh、zsh等。

● 下图为实验室安装好的shell解释器

```
-bash-4.2$ cat /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/bin/tcsh
/bin/csh
/usr/bin/tmux
```

● Windows上的shell

● Power shell

● CMD

● 据说微软是一家非常低调的公司。但却在2006年“大言不惭”地命名和发布了PowerShell。

● 但感觉powershell上手难度比较高，我现在还不怎么会用



SSH client

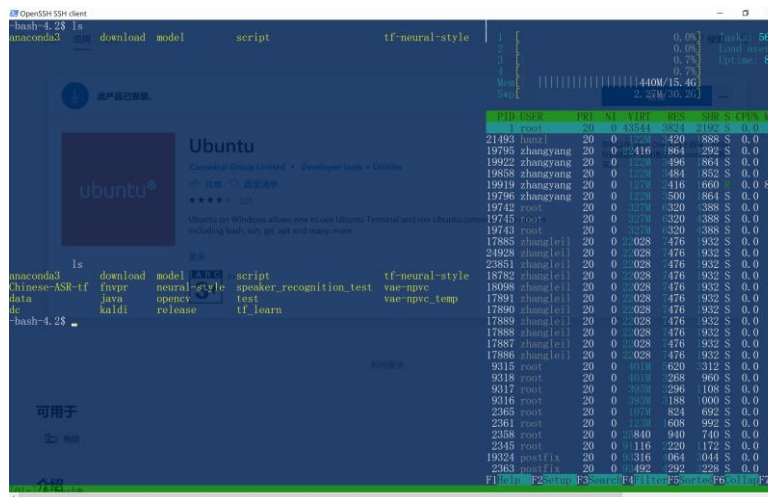
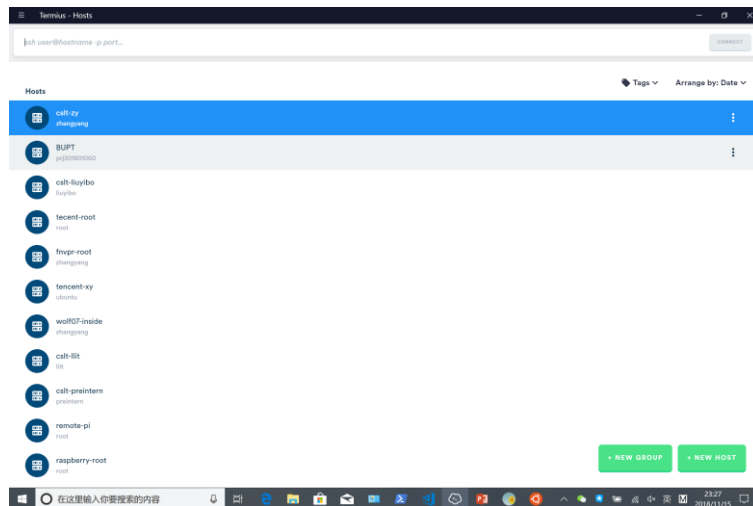
● 我所用过的SSH工具

1. Putty: 感觉不好用

2. Xshell5: 还行

3. Powershell: 安装git以后自带的, 可以使用powershell使用。可以设置透明质感, 比较炫酷

4. Win10应用商店下的工具: 目前我主要使用的SSH工具, 感觉最好用



开源镜像站

● 开源镜像站点

开源镜像站即一个放置开源系统镜像文件的站点.

免费提供镜像文件下载

● 常见的镜像网站

一般开源镜像站的服务器上行带宽都很高，因此用户的下载速度非常快。（由于gwf，或者一些其他因素，官方的下载地址速度不是很快）目前常用的镜像由很多，国内的常用的镜像有清华、中科大、阿里云维护的开源镜像站。



Tsinghua University TUNA Association

📍 Beijing, China

🌐 <https://tuna.moe/>

✉ support@tuna.tsinghua.edu.cn

从上大学开始，我就一直使用清华的开源镜像

在Linux下安装conda

● Conda的分类

- Conda 是一个开源的软件包管理系统和环境管理系统，用于安装多个版本的软件包及其依赖关系，并在它们之间轻松切换。
(感觉和docker是不是也是也有点类似)；简而言之，conda就一个很好用的python工具包，分为miniconda和anaconda。两者我认为基本没有区别，如果网速快，我认为使用miniconda更好一些。(我也不知道这想法对不对)

● 下载 (以清华镜像为例)

AOSP

AUR

CRAN

CTAN

CocoaPods

anaconda

archlinux

archlinuxcn

bananian

bioconductor

centos

chromiumos

cygwin

Anaconda 镜像使用帮助

Anaconda 是一个用于科学计算的 Python 发行版，支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

Anaconda 安装包可以到 <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/> 下载。

TUNA 还提供了 Anaconda 仓库的镜像，运行以下命令：

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/  
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main/  
conda config --set show_channel_urls yes
```

即可添加 Anaconda Python 免费仓库。

运行 `conda install numpy` 测试一下吧。

Miniconda 镜像使用帮助

Miniconda 是一个 Anaconda 的轻量级替代。默认只包含了 python 和 conda，但是可以通过 pip 和 conda 来安装所需要的包。

Miniconda 安装包可以到 <https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/> 下载。

Conda 三方源

当前tuna还维护了一些anaconda三方源。



下载conda

● Conda的下载

- 选择合适的版本，右键点击复制链接，在服务器终端里使用wget + 空格+链接下载

Miniconda3-4.5.4-Windows-x86_64.exe

Miniconda3-latest-Linux-armv7l.sh

Miniconda3-latest-Linux-ppc64le.sh

Miniconda3-latest-Linux-x86.sh

Miniconda3-latest-Linux-x86_64.sh

Miniconda3- 在新标签页中打开

Miniconda3- 在新窗口中打开

Miniconda3- 在新 InPrivate 窗口中打开

Miniconda3- 将目标另存为

Miniconda3- 复制链接

添加到阅读列表

在 百度 中搜索“Miniconda3-latest-Linux-x86_64.sh”

询问 Cortana 关于“Miniconda3-latest-Linux-x86_64.sh”的信息

查看源

● Conda的安装

- 选择合适的版本，右键点击复制链接，在服务器终端里使用wget + 空格+链接下载

54

29

60

56

59

26

41

36

49

52

```
Terminus - Tencent root
-bash-4.2# wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
Terminus - Tencent root
--2018-11-14 20:33:12-- https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Miniconda3-latest-Linux-x86_64.sh
Resolving mirrors.tuna.tsinghua.edu.cn (mirrors.tuna.tsinghua.edu.cn)... 101.6.8.193, 2402:f000:1:408:8100::1
Connecting to mirrors.tuna.tsinghua.edu.cn (mirrors.tuna.tsinghua.edu.cn)|101.6.8.193|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 62574861 (60M) [application/octet-stream]
Saving to: 'Miniconda3-latest-Linux-x86_64.sh'

100%[=====] 62,574,861  2.48MB/s  in 39s

2018-11-14 20:33:51 (1.53 MB/s) - 'Miniconda3-latest-Linux-x86_64.sh' saved [62574861/62574861]

-bash-4.2# ls
ap.py          My-Setting
fnvpr         RPI_web_remote_control-1.1
fnvpr.zip    sql
miniconda3  test
Miniconda3-latest-Linux-x86_64.sh  work
mongodb      work.zip
mongodb-linux-x86_64-rhel70-4.0.4.tgz
-bash-4.2#
```



bashrc & bash_profile

● 使用bashrc个性化定制自己的操作

- Linux认为：用户知道自己想要什么,也明白自己在做什么,并且会为自己的行为负责。
- 实验室默认使用的是bash，配置一个属于自己的bash可以提高自己的工作效率
- Bash_profile：每个用户都可使用该文件输入专用于自己使用的shell信息,当用户登录时,该文件仅仅执行一次!默认情况下,他设置一些环境变量,执行用户的.bashrc文件。
- Bashrc：该文件中包含自己配置的bash信息，当打开新的shell时被读取
- Bash_logout:当每次退出shell时候被执行

```
-bash-4.2$ cat .bashrc
# -----
# .bashrc
# author: zyziszy
# e-mail: zyziszy@foxmail.com
# note: NULL
# -----

# ssh
alias wolf07='ssh wolf07'
alias wolf05='ssh wolf05'
alias wolf07='ssh wolf08'

# untar
alias untar='tar -zxvf'

# Some more alias to avoid making mistakes:
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# anaconda3
export PATH="/work4/zhangyang/anaconda3/bin:$PATH"

# jdk8
export JAVA_HOME=$HOME/java/jdk1.8.0_181
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

# tf
alias tf='source activate tf'

# pytorch
alias pytorch='source activate pytorch'
```



配置环境变量

● 将conda安装路径加入环境变量

- 和window有些不同，Linux使用bashrc来配置环境变量
- 例如，我配置的conda环境变量如下
- `export PATH="/work4/zhangyang/anaconda3/bin:$PATH"`
- 配置完成后需要激活一下
- 使用`source ~/.bashrc`
- 最后使用conda命令测试是否安装成功
- 最新的conda在安装的时候会问你是否加入环境变量，如果选择yes就不用手动添加了

● 不只有bash可以自由定制

- Tmux和vim等一些常用的软件也可以自己定制，这是我在备份GitHub上的一些配置

The screenshot shows a GitHub repository interface for 'My Setting backup'. At the top, there are navigation links for 'settings-sync', 'settings', and 'Manage topics'. Below that, repository statistics are shown: 28 commits, 1 branch, 0 releases, 1 contributor, and MIT license. A table of commit history is visible, listing files like conda, git, jetbrain, linux, vim, vscode, win10, .gitignore, LICENSE, README.md, run.bat, run.py, and run.sh with their respective commit dates. At the bottom, the README.md file is open, displaying the title 'MY-Setting' and the text 'back up my personal settings in this repo.'

File	Commit Type	Time Ago
conda	update	2 days ago
git	add git_config.sh	28 days ago
jetbrain	commit	a month ago
linux	add .dir_colors	8 days ago
vim	update vimrc	a month ago
vscode	add vscode python launch.json	27 days ago
win10	update	a month ago
.gitignore	commit	a month ago
LICENSE	Initial commit	a month ago
README.md	update	a month ago
run.bat	commit	a month ago
run.py	commit	a month ago
run.sh	commit	a month ago

使用Conda创建python环境

- `Conda create -n tf python=3.6`
- `Conda remove -n tf -all -y`
- `source activate tf`
- `Source deactivate tf`
- `Conda env list`
- `Conda -V`
- `Conda env export > environment.yml`
- `conda env create -f environment.yml`

在wolf集群快速使用conda

- 可以直接使用我配置好的tf和torch环境

在bashrc里加入如下配置

```
# anaconda3
export PATH="/work4/zhangyang/anaconda3/bin:$PATH"
```

```
# tf
alias tf='source activate tf'
```

```
# pytorch
alias pytorch='source activate pytorch'
```

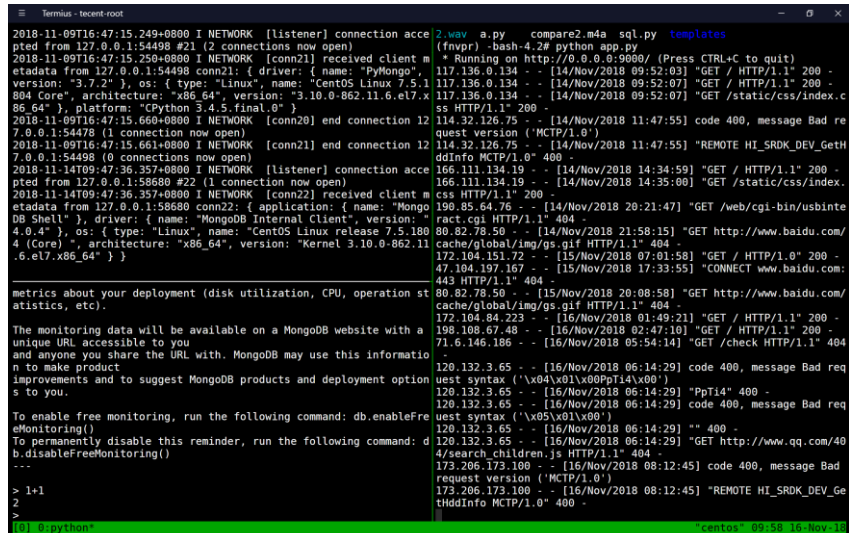
注意在bash_profile得设置好启动终端时运行一遍bashrc



我与TMUX和VIM

● Tmux

- 在一开始，我一直觉得Tmux是很难用的
- 但有次吃饭的时候，杜文强学长告诉我这个工具非常方便，于是回到学校我记了了几个指令，之后就深深的喜欢上了这个小工具。



● vim

- Vim应该算是上古神器了，我在小学就听说过这个，之前一直以为是黑客用的破解软件。是上了大学才知道这原来是个文本编辑器。。。
- Ps: Vim和emacs哪个更好用一直争论不休。。。
- 在我还没进入实验室的时候，就一直在记vim指令，但后面我在实验室配置的vimrc被阿汤哥全部删了，之后就放弃使用vim了（还是微软的vscode好用）

```

1 #!/usr/bin/env python
2 # coding=utf-8
3
4 # *****
5 # > File Name: ap.py
6 # > Author: Yang Zhang
7 # > Mail: zyziszy@foxmail.com
8 # > Created Time: Fri 09 Nov 2018 04:53:02 PM CST
9 # *****
10
11 # enroll the singer vector to database
12
13
14 # import this in Linux with engine lib
15 try:
16     from fnvpr import sre
17 except:
18     print("import sre failed")
19
20 from pymongo import MongoClient
21 import os
22
23 if __name__ == '__main__':
24
25     # connect to database
26     client = MongoClient('mongodb://localhost:27017/')
27     db = client["fnvpr-database"]
28
29     # singer vector collection
30     singer_vector_col = db['singer-vector-collection']
31
32     # songs feature collection
33     songs_feature_col = db['songs-feature-collection']
34
35     singer_name = input("input singer name:")
36     singer_gender = input("input singer gender:")
37     singer_wav = input("input singer wav path:")
38
39     while os.path.exists(singer_wav) == False:
40         singer_wav = input("input singer wav path:")
41
42     ap.py 77L, 2051C

```

我认为还不错的Linux教程

- 《鸟哥的Linux私房菜》
这可能是目前我唯一读过和Linux有关的书，感觉还不错。因为没看过太多，使用也不知道还有没有更好的书推荐
- 知乎上的专栏“Linux中国”
这个专栏会分享一些Linux的新闻和使用技巧，感觉很不错，但有些文章写的比较难
- 自己购买一个服务器
我再腾讯云上有自己的vps，刚买服务器那个假期我记住了很多命令
- 刷PT逛论坛
用Linux服务器做种，刷流量~
- 《Just for fun》
Linus写的自传

Windows下远程调试Linux下的python程序

- 使用vim
这个学习成本根据有点高，至今我也还不太会用。但我目前主要用这个
- pycharm
这个挺方便的，但有跳板机的存在可能需要连接内网，连接到指定的wolf机上进行。外网调试不是很方便
- 使用jupyter notebook
由于跳板机的存在，权限不足等问题。我暂时没用弄成功，但应该这个也是一个很好的方法
- 使用vscode
Vscode里有很多sftp的插件，非常好用，并且可以解决跳板机的问题

谢谢观看

