

UNIX command line

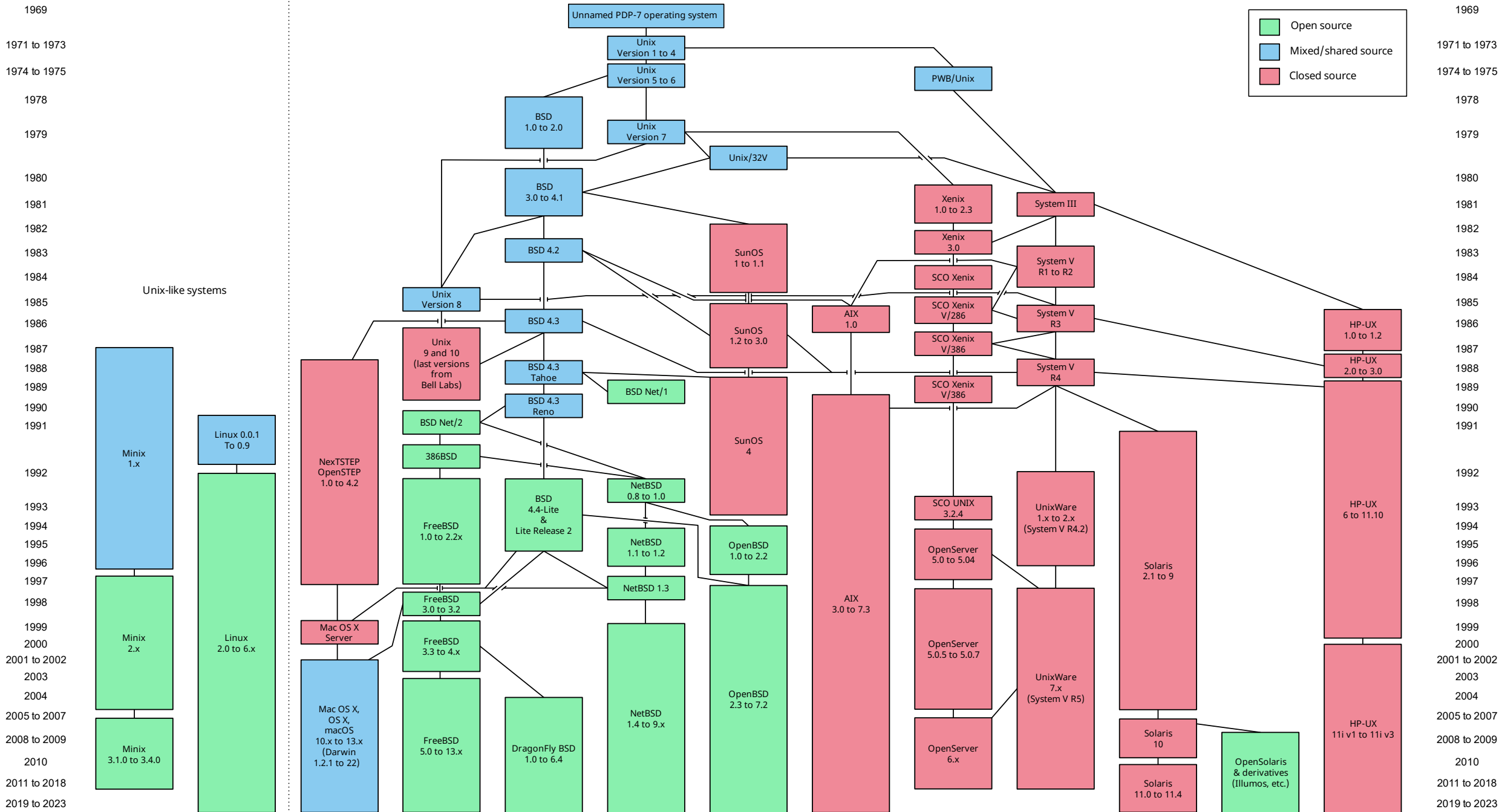
история, принципы, понятия

Аспирант АБИБ ЮФУ
2 года обучения
Дёмин К. А.

Ростов-на-Дону, 2024

Bell labs и компьютерные сети





Философия UNIX

- пишите программы, которые делают что-то одно и делают это хорошо;
- пишите программы, которые бы работали вместе;
- пишите программы, которые бы поддерживали текстовые потоки, поскольку это универсальный интерфейс.

Другие особенности UNIX

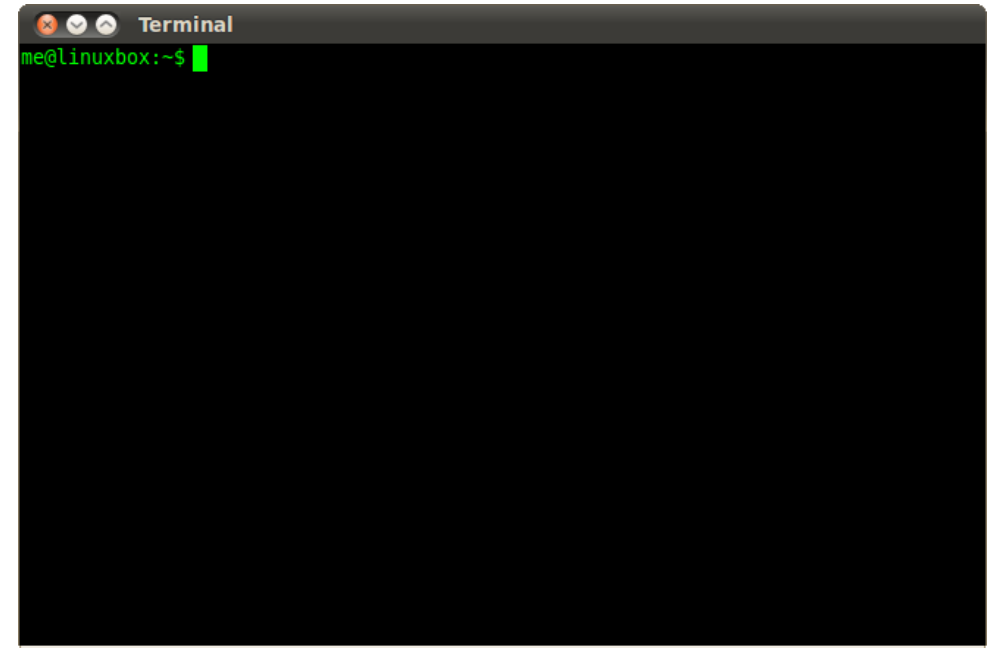
- использование **простых текстовых файлов** для настройки и управления системой;
- широкое применение утилит, запускаемых из **командной строки**;
- взаимодействие с пользователем посредством виртуального устройства — **терминала**;
- представление физических и виртуальных устройств и некоторых средств межпроцессного взаимодействия в виде **файлов**;
- использование конвейеров из нескольких программ, каждая из которых выполняет одну задачу (**пайплайн**).

Почему CLI вместо GUI

1. Очень просто
2. Гибко
3. Модульно
4. Репрезентативно
5. Почти бесплатно
6. Крайне эффективно для сложных вычислительных задач (сборка геномов, выравнивание последовательностей)
7. Не требует больших *локальных* вычислительных мощностей или высокой пропускной способности интернет соединения

Понятия, которые необходимо изучить

- CLI, Terminal, Linux shell
- Директории, абсолютный и относительный путь
- Стандартный ввод, стандартный вывод, стандартный вывод ошибок
- Анатомия командной строки
- Пайплайн
- Wildcards



Анатомия командной строки

1

```
$ cat -b file1.txt file2.txt > file3.txt  
$ samtools sort -l --threads 10 -i file.bam -o sorted_file.bam
```

2

```
$ cat -b file1.txt file2.txt > file3.txt  
$ samtools sort -l --threads 10 -i file.bam -o sorted_file.bam
```

3

1 приглашающий символ (промпт)

2 вызов программы

3 флаг

4 подкоманда программы

5 длинный флаг / длинноформатный параметр

6 значение параметра

7 перенаправление стандартного вывода

} Параметры

▼ `cat(1)` `-b` `file1.txt` `file2.txt` `>` `file3.txt`

concatenate files and print on the standard output

`-b, --number-nonblank`
number nonempty output lines, overrides `-n`

Concatenate FILE(s), or standard input, to standard output.

With no FILE, or when FILE is `-`, read standard input.

Before a command is executed, its input and output may be redirected using a special notation interpreted by the shell. Redirection may also be used to open and close files for the current shell execution environment. The following redirection operators may precede or appear anywhere within a simple command or may follow a command. Redirections are processed in the order they appear, from left to right.

Redirecting Output

Redirection of output causes the file whose name results from the expansion of word to be opened for writing on file descriptor n, or the standard output (file descriptor 1) if n is not specified. If the file does not exist it is created; if it does exist it is truncated to zero size.

The general format for redirecting output is:

`[n]>word`

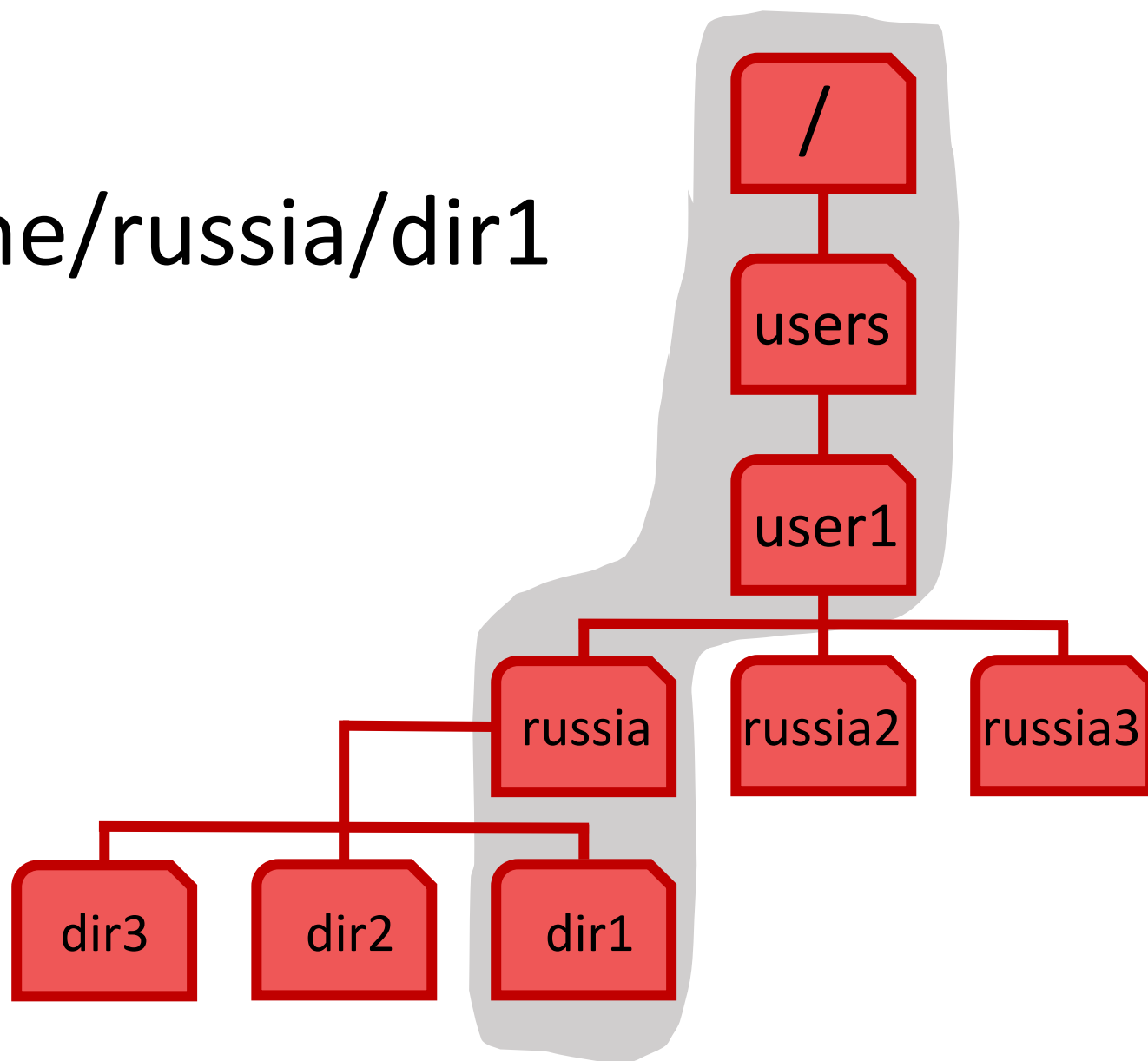
If the redirection operator is `>`, and the `noclobber` option to the `set` builtin has been enabled, the redirection will fail if the file whose name results from the expansion of word exists and is a regular file. If the redirection operator is `>|`, or the redirection operator is `>` and the `noclobber` option to the `set` builtin command is not enabled, the redirection is attempted even if the file named by word exists.

Директории и пути

1. /users/user1/home/russia/dir1

2. russia/dir1

3. ../../russia/dir1



Стандартный ввод и вывод

STDIN	<	прочитать из файла или ввода пользователя
STDOUT	>	записать стандартный вывод в файл
STDOUT	>>	добавить стандартный вывод к файлу
STDERR	2>	вывести отчет об ошибках в файл
PIPE		перенаправить STDOUT в STDIN

```
$ cat -b file1.txt file2.txt > file3.txt
```

Пайплайны

```
$ history | grep -w cat | sort > sorted history.txt
```

Wildcards (символы подстановки, «джокеры»)

```
cat *.txt > new.txt
```

```
cat file*.txt > new.txt
```

```
cat file?.txt > new.txt
```

Некоторые полезные команды

- **ls** показать содержимое
- **cd** переместиться
- **mv** переместить
- **cp** копировать
- **mkdir** создать директорию
- **cat** манипулировать текстом
- **rm** удалить
- **man** показать мануал
- **nano** открыть редактор
- **history** показать историю
- **grep** найти паттерн
- **sudo** «сделать как» администратор
- **wget** скачать из интернета
- **pwd** показать текущий путь
- **tar** разархивировать

Дополнительные материалы

- <https://github.com/konstantin-demin/metagenomics-lib>
- <https://explainshell.com/>
- <https://www.youtube.com/watch?v=nnLI9N5Q0jc>