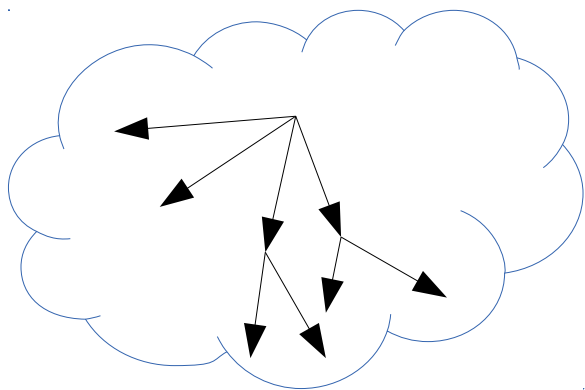


Файлы

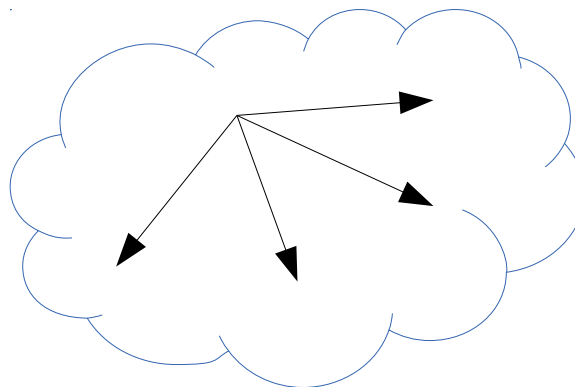
Unix это...

- Процессы:
 - «тяжелые»
 - «легкие»
- Файлы:
 - Файловая система
 - Файлы со стороны процессов (потoki, дескрипторы)
- Взаимодействие процессов (IPC):
 - Локальное
 - Сетевое

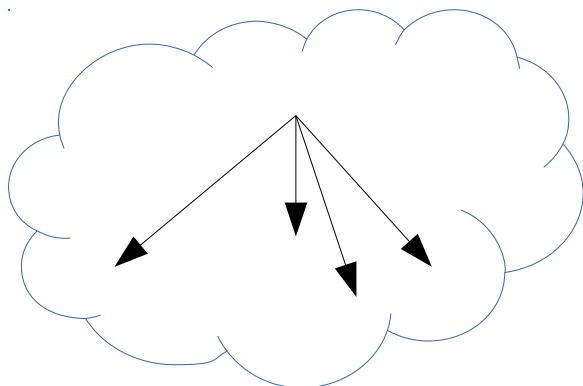
C:



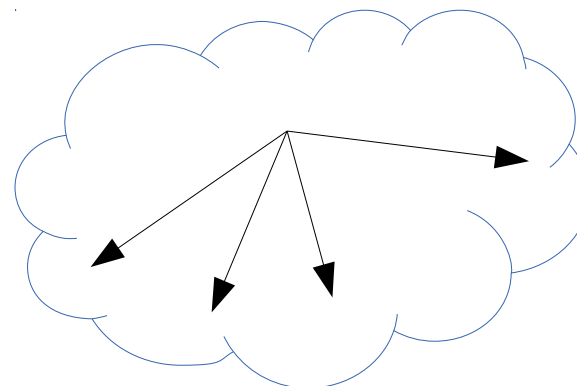
E:



D:

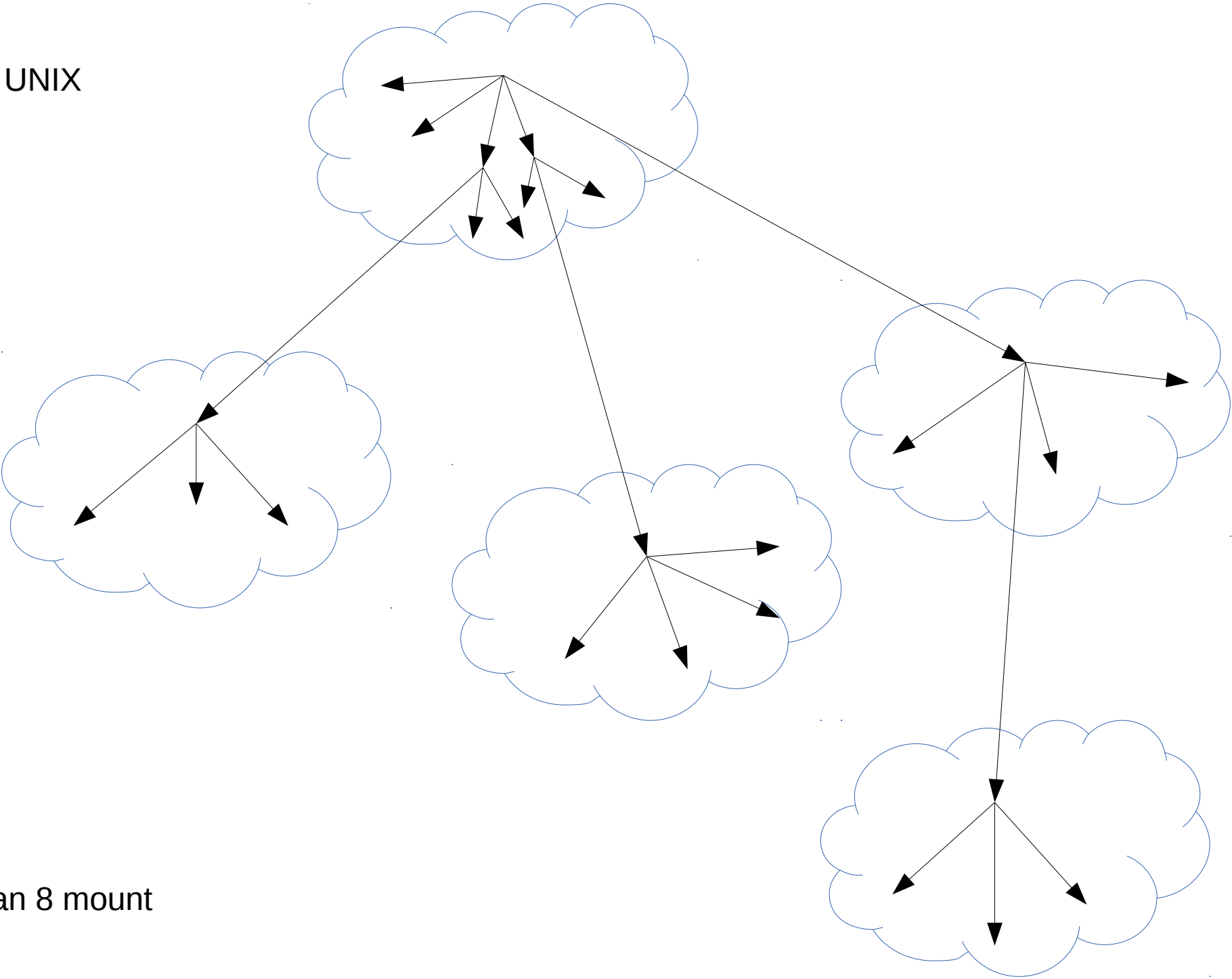


F:



Файловые системы в Windows

B UNIX



man 8 mount

Пример вывода mount (Linux)

- /dev/sda8 on / type ext4 (rw,errors=remount-ro)
 - proc on /proc type proc (rw,noexec,nosuid,nodev)
 - sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
 - none on /sys/fs/fuse/connections type fusectl (rw)
 - none on /sys/kernel/debug type debugfs (rw)
 - none on /sys/kernel/security type securityfs (rw)
 - udev on /dev type devtmpfs (rw,mode=0755)
 - devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
 - tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
 - none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
 - none on /run/shm type tmpfs (rw,nosuid,nodev)
- /dev/sda6 on /home type ext4 (rw)

Партиции

- Linux:
 - /dev/sda — первый диск, /dev/sda1 — первая партиция, sda2 — вторая и т. д.
 - Второй диск: sdb, партиции: sdb1, sdb2, ...
- FreeBSD:
 - 1 SATA/IDE диск: /dev/ada0, /dev/ada0s1 — первая партиция, s2 — вторая и т. д.
 - Второй диск: ada1
- MAC OS X: disk0s1, disk1s3, ...



Файловая система (man 7 hier)

- Корень (/)
 - /bin — исполнимые файлы (нужны для ОС)
 - /lib
 - /etc — настройки, скрипты запуска, ...
 - /etc/fstab — таблица монтирования ФС
 - /etc/passwd, /etc/group — информация о пользователях
 - /tmp — временное хранилище
 - /usr — файлы для пользователя, включая:
 - /usr/bin — исполнимые файлы (например, редакторы)
 - /usr/include — файлы для компилирования C/C++
 - /dev — файлы устройств

Файлы

Именованная область данных.

С точки зрения

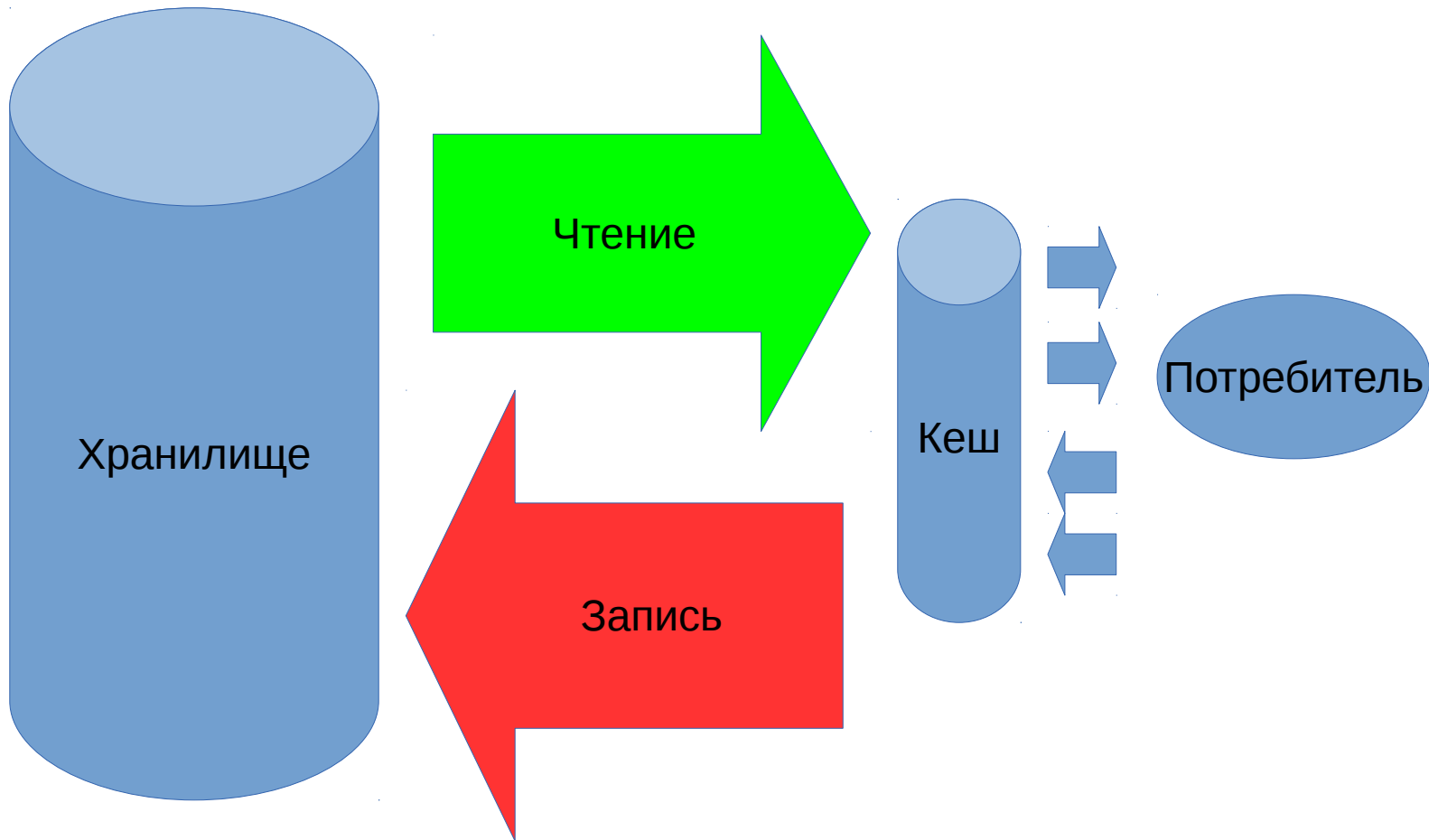
- Администратора:
 - /bin/mount
 - /etc/fstab
- Пользователя:
 - ls, cat, vi, test, sort, ...
- Программиста

Файловые дескрипторы

- Получение: `open(char *path, int flags, ...)`
- Работа:
 - Чтение: `read(int d, void *buf, size_t nbytes)`
 - Запись: `write(int d, void *buf, size_t nbytes)`
 - Перемещение: `lseek(int d, offset, int whence)`
- Завершение работы: `close(int d)`

Системный вызов — это долго и дорого.

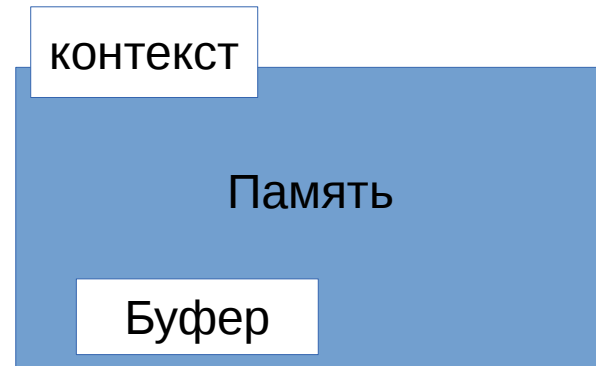
Кеширование



Банк ↔ банкомат ↔ пользователь

Буферизованный ввод/вывод

- FILE *
- open: fopen
- read: fscanf/scanf, fgetc/getchar, fread
- write: fprintf/printf, fputc/putchar, fwrite
- lseek: fseek
- close: fclose



man 3 <имя функции>

Соответствие режимов

- "r": O_RDONLY
- "w": O_WRONLY, O_CREAT, O_TRUNC
- "a": O_WRONLY, O_CREAT, O_APPEND
- "r+": O_RDWR
- "w+": O_RDWR, O_CREAT, O_TRUNC
- "a+": O_RDWR, O_CREAT, O_APPEND

Пример: сколько * будет?

```
void main(){  
    putchar('*');  
    fork();  
    putchar('*');  
}
```

Пример: сколько * будет?

```
void main(){  
    putchar('*');  
    fork();  
    putchar('*');  
    _exit(1);  
}
```

Работа с директориями

- Получение:
 - `DIR *opendir(const char *name);`
 - `DIR *fdopendir(int fd);`
- Работа:
 - `struct dirent *readdir(DIR *dirp);`
 - `int readdir_r(DIR *dirp, struct dirent *entry, struct dirent **result);`
- Завершение работы: `int closedir(DIR *dirp);`

man 3

Пример opendir/readdir

```
dirp = opendir(".");
if (dirp == NULL)
    return (ERROR);
len = strlen(name);
while ((dp = readdir(dirp)) != NULL) {
    if (dp->d_namlen == len && strcmp(dp->d_name, name) == 0) {
        (void)closedir(dirp);
        return (FOUND);
    }
}
(void)closedir(dirp);
return (NOT_FOUND);
```


Дескрипторы и память

```
void *mmap(void *addr, size_t length, int prot, int  
flags, int fd, off_t offset);
```

Зачем:

- Работа с файлом не через read/write/lseek, а через `a[i]+=10;`
- Разделяемая память (через файл или специальный объект)

a[20]+=10 через read/write

```
int fd;  
int a_temp;  
fd=open("int-array.data", O_RDWR);  
// проверили fd, файл бинарных int-значений  
lseek(fd, 20*sizeof(a_temp), SEEK_SET);  
read(fd,&a_temp,sizeof(a_temp));  
a_temp+=10;  
lseek(fd, 20*sizeof(a_temp), SEEK_SET);  
write(fd,&a_temp,sizeof(a_temp));
```