УПРАВЛЕНИЕ ДИРЕКТОРИЯМИ

Системные вызовы и библиотеки Unix SVR4

Иртегов Д.В.

ФФ/ФИТ НГУ

Электронный лекционный курс подготовлен в рамках реализации

Программы развития НИУ-НГУ на 2009-2018 г.г.

СВОЙСТВА ДИРЕКТОРИИ

- используется для организации обычных файлов, программных каналов, специальных файлов и других директорий
- формат, требуемый операционной системой
- содержит имена файлов и inode-номера
- записи в директории называются связями
- нет ограничения глубины вложенности поддиректорий
- права доступа
  - r: право на чтение
  - w: право на запись
  - x: право на поиск

Системно-независимая структура записи в директории

```
struct dirent {
  ino_t d_ino;
   /* "inode number" of entry */
  off_t d_off;
  /* offset of disk directory entry */
  unsigned short d_reclen;
  /* length of this record */
  char d_name[1]; /* name of file */
};
```

Изменение текущей директории

ИСПОЛЬЗОВАНИЕ

```
#include <unistd.h>
int chdir(conts char *path);
int fchdir(int fildes);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

- успех - 0
- неуспех - -1 и errno установлена

Создание директории

ИСПОЛЬЗОВАНИЕ

```
#include <sys/types.h>
#include <sys/stat.h>
int mkdir(const char *path,
      int mode);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

Удаление директории

ИСПОЛЬЗОВАНИЕ

```
#include <unistd.h>
int rmdir(conts char *path);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

Создание и удаление цепочки директорий

ИСПОЛЬЗОВАНИЕ

```
#include <unistd.h>
int mkdirp(conts char *path,
       int mode);
int rmdirp(char *path, char *path1);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - mkdirp -1 и errno установлена

rmdirp - -1, -2 или -3

Чтение директории

ИСПОЛЬЗОВАНИЕ

```
#include <dirent.h>
DIR *opendir(const char *filename);
struct dirent *readdir(DIR *dirp);
long teldir(DIR *dirp);
void seekdir(DIR *dirp, long loc);
void rewinddir(DIR *dirp);
int closedir(DIR *dirp);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех – opendir, readdir - указатель на структуру

closedir - 0

telldir - текущая позиция

неуспех - opendir, readdir - NULL и errno установлена

telldir, closedir - -1 и errno установлена

Чтение директории

readdir возвращает один и тот же указатель, поэтому память из-под struct dirent * освобождать не надо

В действительности, структура DIR содержит небольшой кольцевой буфер

Есть форма

```
int readdir_r(DIR *restrict dirp,
       struct dirent *restrict entry,
        struct dirent **restrict result);
```

При размещении struct dirent необходимо также разместить память под строку d_name;

размер этой строки рекомендуется определять при помощи `pathconf(dir, _PC_NAME_MAX)`, т.е. динамически

Создание связи

ИСПОЛЬЗОВАНИЕ

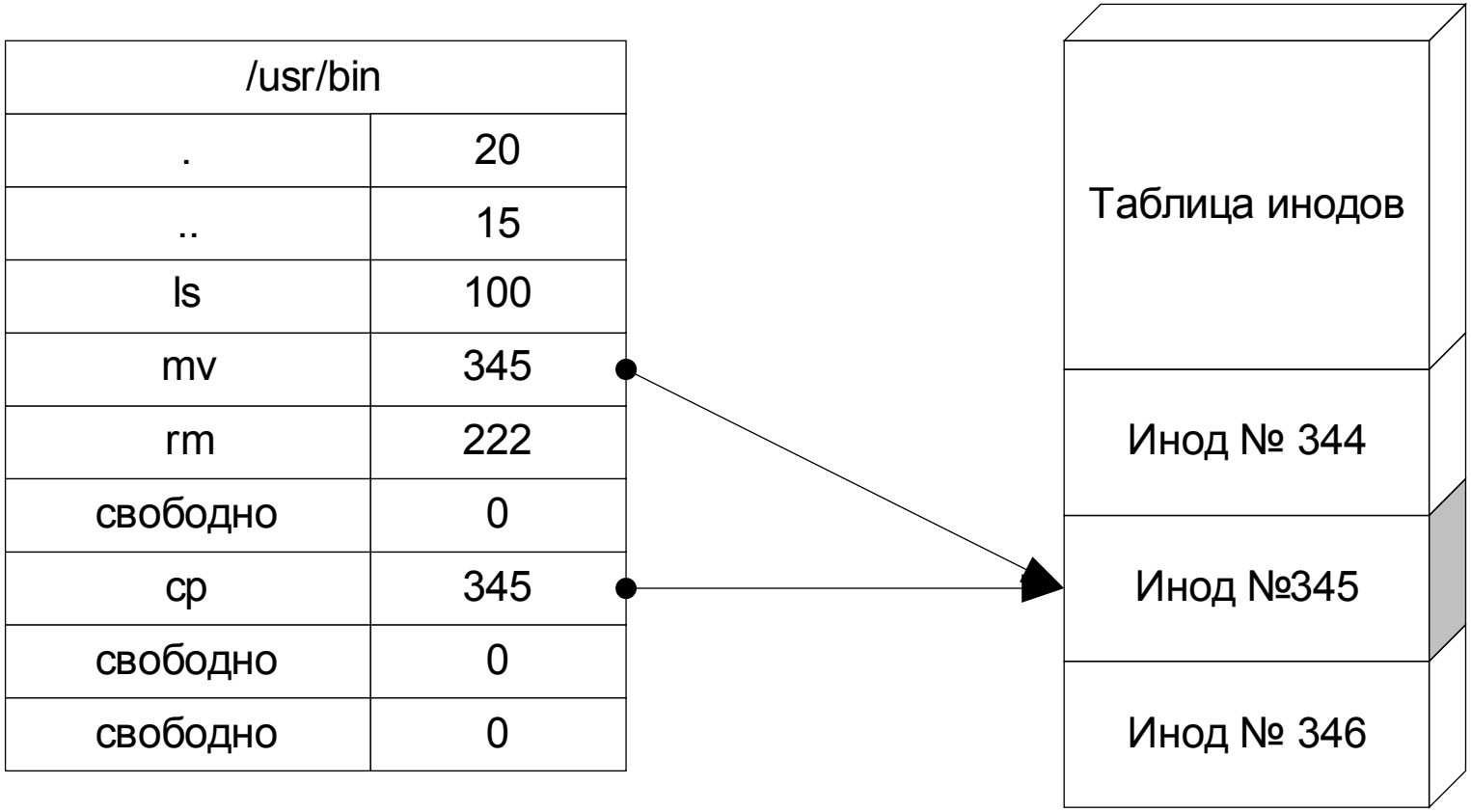
```
#include <unistd.h>
int link(const char *path1,
  const char *path2);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

Жесткая связь



Создание символической связи

ИСПОЛЬЗОВАНИЕ

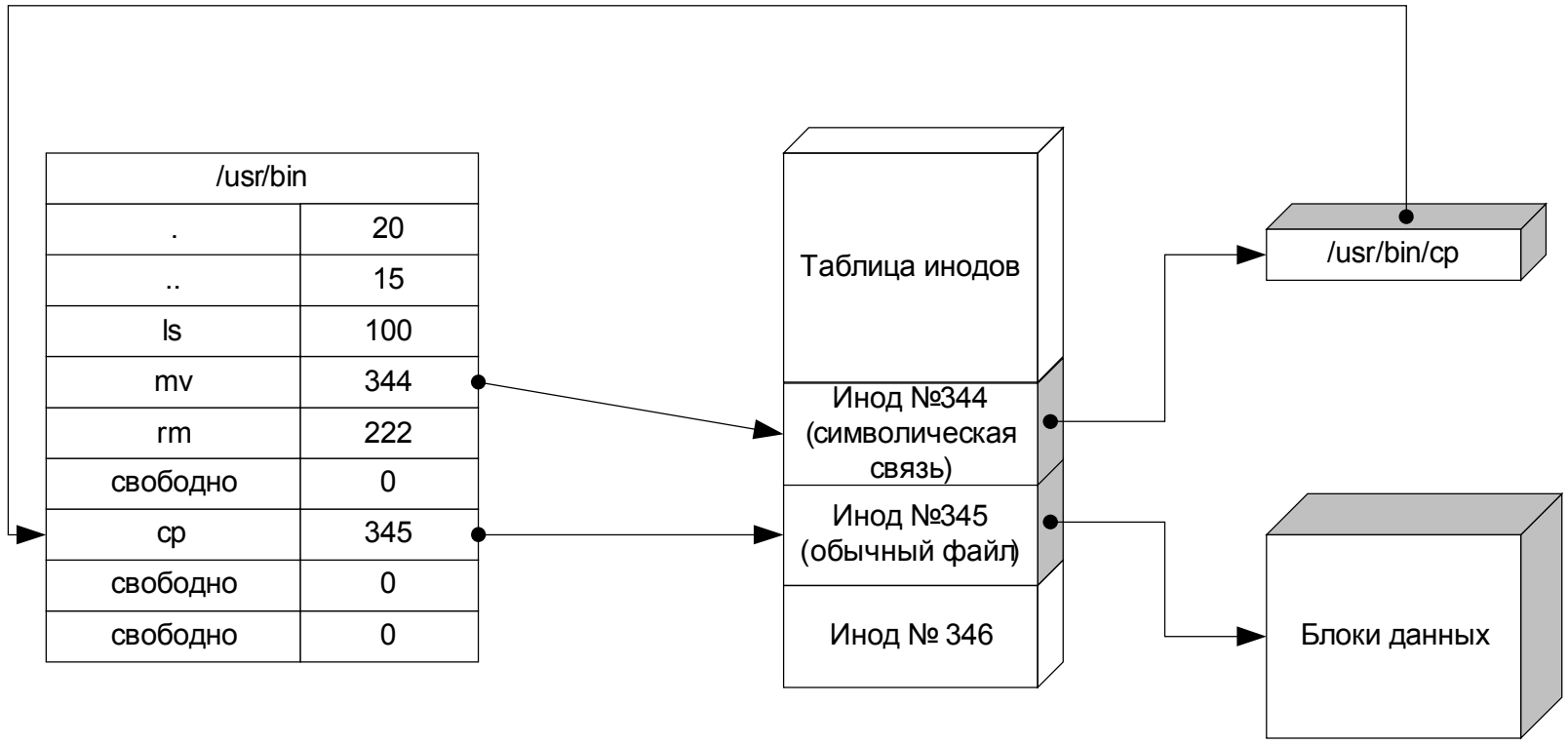
```
#include <unistd.h>
int symlink (const char *name1,
  const char *name2);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

Символическая связь



Чтение значения симлинка

ИСПОЛЬЗОВАНИЕ

```
#include <unistd.h>
int readlink(const char *path,
  void *buf, size_t bufsize);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - количество символов, считанных в буфер

неуспех - -1 и errno установлена

Следование симлинкам

следуют символическим связям	`open(2)`
	`chmod(2)`
	`chown(2)`
	`chgrp(2)`
	`chdir(2)`
	`stat(2)`

не следуют символическим связям	`lchown(2)`
	`readlink(2)`
	`lstat(2)`
	`link(2)`
	`unlink(2)`
	`rename(2)`
	`rmdir(2)`

Удаление записи из директории

ИСПОЛЬЗОВАНИЕ

```
#include <unistd.h>
int unlink(const char *path);
int remove(
    const char *filename);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

Переименование файла

ИСПОЛЬЗОВАНИЕ

```
#include <stdio.h>
int rename(const char *old,
    const char *new);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех -1 и errno установлена

Разбор путевого имени

ИСПОЛЬЗОВАНИЕ

```
cc -lgen
#include <libgen.h>
char *dirname(char *path);
char *basename(char *path);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех – компонент имени

неуспех - NULL