Семинар 13. Файлы

Файловые дескрипторы

- Неотрицательное целое число
- Когда создается новый поток ввода-вывода, ядро возвращает процессу, создавшему поток ввода-вывода, его файловый дескриптор
- По умолчанию 0 - **stdin**, 1 - **stdout**, 2 - **stderr**. Это (необязательное) соглашение оболочек, но не составная часть ядра
- Стандарт POSIX - **STDIN_FILENO**, **STDOUT_FILENO** и **STDERR_FILENO**

Файловые дескрипторы

- Под файловые дескрипторы отводится диапазон чисел от 0 до **OPEN_MAX**
- Посмотреть значение **OPEN_MAX** можно с помощью **sysconf(_SC_OPEN_MAX)**;
- OPEN_MAX не может быть меньше, чем **_POSIX_OPEN_MAX**
- Увеличить можно с помощью команды **ulimit –n**

Open

- Заголовочный файл **fcntl.h**
- **int open(const char *pathname, int oflag, ... /* mode_t mode*/);**
- Возвращает дескриптор в случае успеха, -1 в случае ошибки
- (обязательно) **O_RDONLY** или **O_WRONLY** или **O_RDWR**
- (необязательно)**O_APPEND**, **O_CREAT**, **O_EXCL**, **O_TRUNC**, **O_NOCTTY**, **O_NONBLOCK**, **O_DSYNC**, **O_RSYNC**, **O_SYNC**
- **mode** нужен только при создании файла, список констант

Creat

- Заголовочный файл **fcntl.h**
- **int creat(const char *pathname, mode_t mode);**
- Возвращает дескриптор, доступный только для записи, или -1 в случае ошибки
- Эквивалентна open(pathname, **O_WRONLY | O_CREAT | O_TRUNC**, mode);

Close

- Заголовочный файл **unistd.h**
- **int close(int filedes);**
- Возвращает 0 в случае успеха, -1 в случае ошибки
- При завершении процесса все открытые им файлы автоматически закрываются

Lseek

- Заголовочный файл **unistd.h**
- **off_t lseek(int filedes, off_t offset, int whence);**
- Возвращает новую текущую позицию файла в случае успеха, –1 в случае ошибки
- **SEEK_SET** - смещение от начала файла, **SEEK_CUR** - смещение от текущей позиции, **SEEK_END** - смещение от конца файла
- Передав 0, можно узнать текущее смещение и есть ли возможность свободного перемещения

Read

- Заголовочный файл **unistd.h**
- **ssize_t read(int filedes, void *buf, size_t nbytes);**
- Возвращает количество прочитанных байт, 0 в случае конца файла, -1 в случае ошибки
- В случае успеха текущая позиция будет увеличена
- Важно помнить, что по ряду причин считано может быть меньше байт, чем запрошено

Write

- Заголовочный файл **unistd.h**
- **ssize_t write(int filedes, const void *buf, size_t nbytes);**
- Возвращает количество записанных байт, -1 в случае ошибки
- Важно помнить, что по ряду причин записано может быть меньше байт, чем передано