

# Лекция 34

## Обзор средств синхронизации и межпроцессной коммуникации

# Рассмотренные

- Файловая система, файлы
- Блочные и символьные устройства
- Разделяемая память (файлы, отображаемые в память)
- Pipe
- Fifo
- Socket
- Signals

# Таймер

- Отмеряют интервалы времени и уведомляют процесс о срабатывании
- Самые простые: `alarm(sec)`, `ualarm(usec)` – при срабатывании `SIGALRM`, однократно
- `setitimer/getitimer` – выбор типа времени, выбор однократно/периодически
  - `ITIMER_REAL` – реальное время – `SIGALRM`
  - `ITIMER_VIRTUAL` – время пользовательского режима – `SIGVTALRM`
  - `ITIMER_PROF` – время пользовательского режима и ядра (профилирование) - `SIGPROF`

# POSIX timers

- `timer_create`, `timer_settime`, `timer_gettime`, `timer_delete`
- Несколько таймеров на один процесс/нить
- При срабатывании: ничего не делать, генерировать сигнал, запускать функцию в НИТИ

# Задание интервалов

```
struct itimerval {
    struct timeval it_interval; /* интервал до 2, 3, ... срабатывания */
    struct timeval it_value;    /* интервал до первого срабатывания */
};
struct timeval {
    time_t      tv_sec;          /* seconds */
    suseconds_t tv_usec;        /* microseconds */
};

int setitimer(int which,
              const struct itimerval *new_value,
              struct itimerval *old_value);
```

# Timerfd (Linux)

- Отображает таймеры на файловые дескрипторы
- `timerfd_create` – создание таймера
- `timerfd_settime` – установка
- `timerfd_gettime` – получение состояния
- `Read` – ждать срабатывания

# Семафоры

- Семафор – базовый низкоуровневый примитив синхронизации
- Значение –  $s$  – целое значение  $\geq 0$
- Две операции:  $s.up(value)$ ,  $s.down(value)$ 
  - down: атомарная операция:  
if ( $s \geq value$ )  $value -= s$ ;  
else  $sleep\_on(s)$ ;
  - up: атомарная операция:  
 $s += value$ ;  $wake\_up(s)$ ;

# Семафоры на уровне системных ВЫЗОВОВ

- SysV IPC: (semget, semop, semctl) – legacy
- Posix Sem: (sem\_init, sem\_post, sem\_wait)
- С помощью pipe или fifo: read/write – используется файловый дескриптор
- Eventfd - “легковесный” семафор на файловых дескрипторах: eventfd



# Очереди сообщений

- Очереди сообщений сохраняют границы сообщений
- System V IPC: (semget) – legacy
- POSIX messages: (mq\_open, mq\_receive, mq\_send)
- mq\_open возвращает тип mqd\_t – но на самом деле это файловый дескриптор (в Linux)

# Другие использования файловых дескрипторов (Linux)

- Inotify — отслеживание изменений в файловой системе (`inotify_init`)
- Fanotify – отслеживание изменений в файловой системе (`fanotify_init`)
- Memfd – анонимные временные файлы (`memfd_create`)
  - В частности, можно делать (`flexecve`)
- Pidfd – файловые дескрипторы для процессов (`pidfd_open` – Linux 5.3)
- Псевдотерминалы (`getpt` - будут рассмотрены)