Семинар 3

Processes

Насущное

- Github - как отписаться от уведомлений
- commit messages!!!!!
- Домашка

Syscalls for file management

fd = open(path, how)

s = close(fd)

n = read(fd, buff, nbytes)

n = write(fd, buff, nbytes)

position = lseek(fd, offset, whence)

s = stat(name, &buf)

Syscalls for filesys/dirs management

s = mkdir(name, mode)

s = rmdir(name)

s = link(name1, name2)

s = unlink(name)

s = chdir(dirname)

s = mount(special, name, flag)

s = unmount(special)

Operating system structure

- Monolithic
- Layered
- Microkernel
- Client-Server model
- VMs
- Exokernel

Processes again

born again

fork again

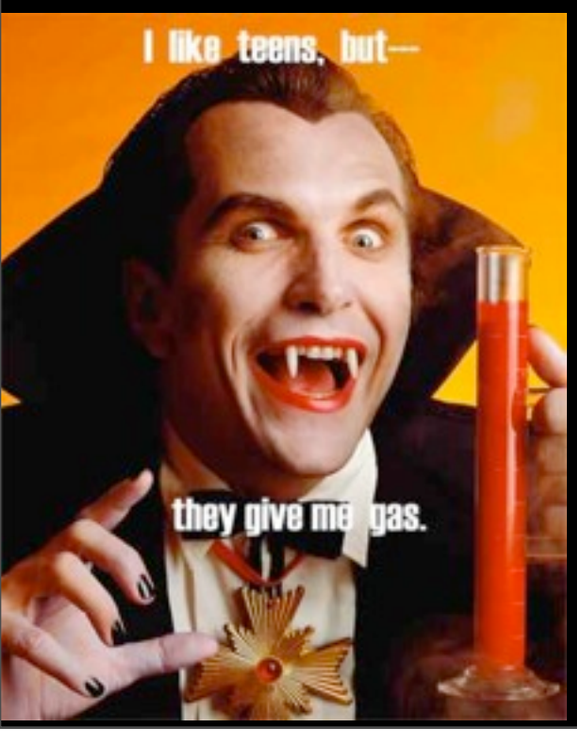
Process

- program? task?
- pseudoparallelism once again
- UID, GID
- PID, PPID
- address space, process table

Process creation

pid = fork()

daemons



System initialization
Execution of a process creation system call
User request to create new process
Initialization of batch job

Process termination

Normal exit
Error exit
Fatal error
Killed

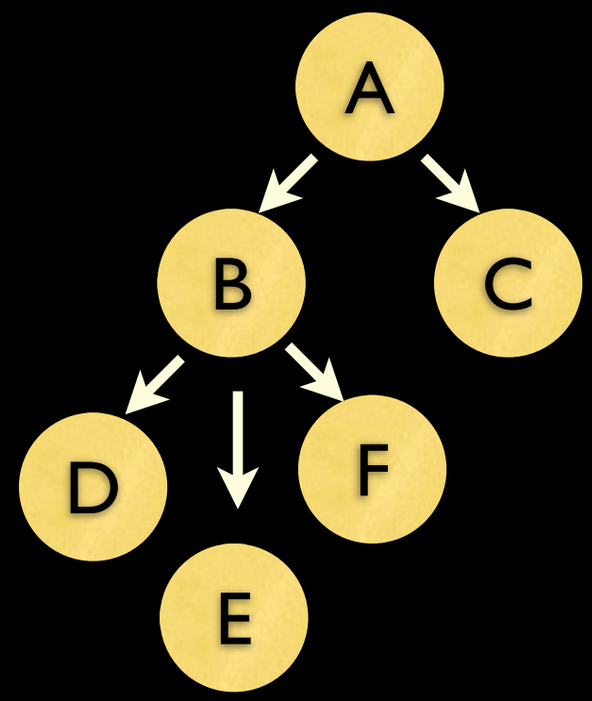
top

kill

s = kill(pid, signal)

Process hierarchies

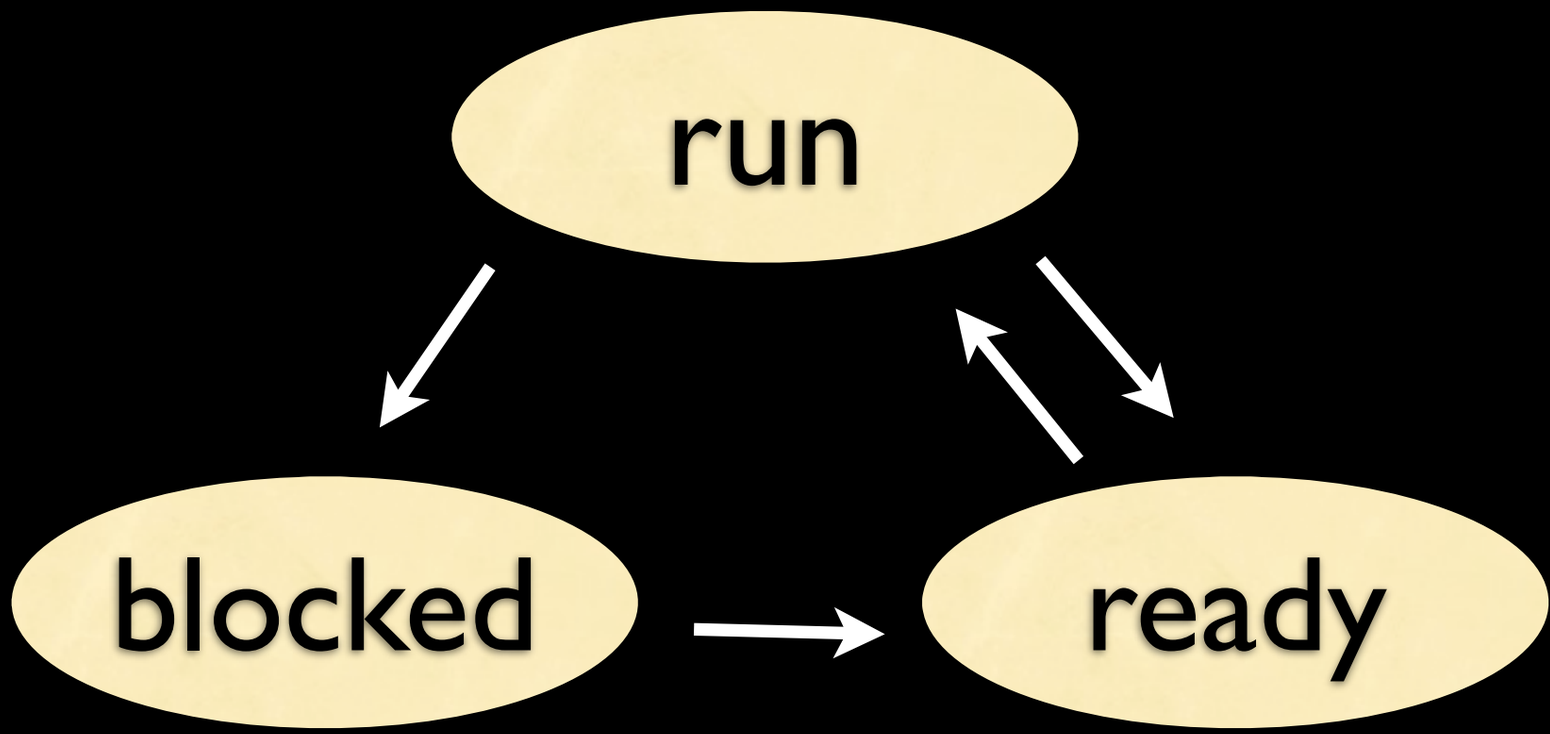
```
if(fork()==0){
    // child stuff
} else {
    // parent stuff
}
```



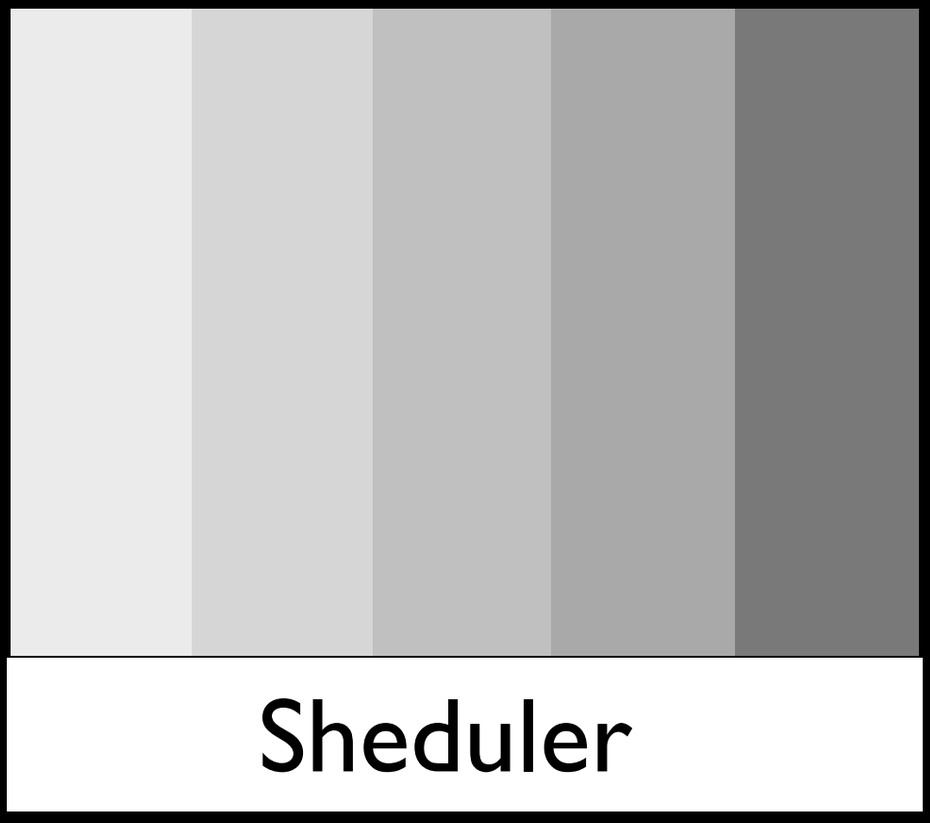

```
pid = waitpid(pid, &static, options)
```

```
s = execve(name, argv, environp)
```

Simple process states



Implementation of Processes



Process management

Registers
Program counter
Program status word
Stack pointer
Process state
Priority
Shedul params
pid, ppid
signals
time started
CPU time used
Children' CPU time
time to next alarm

File management

Root directory
Working directory
File deskriptors
User ID
Group ID

Memory management

Pointer to text segment info
Pointer to data segment info
Pointer to stack segment info

Modeling multiprogramming

Задача I из задания I

- комментарии...
- input file format

fork bomb: while(1) fork();