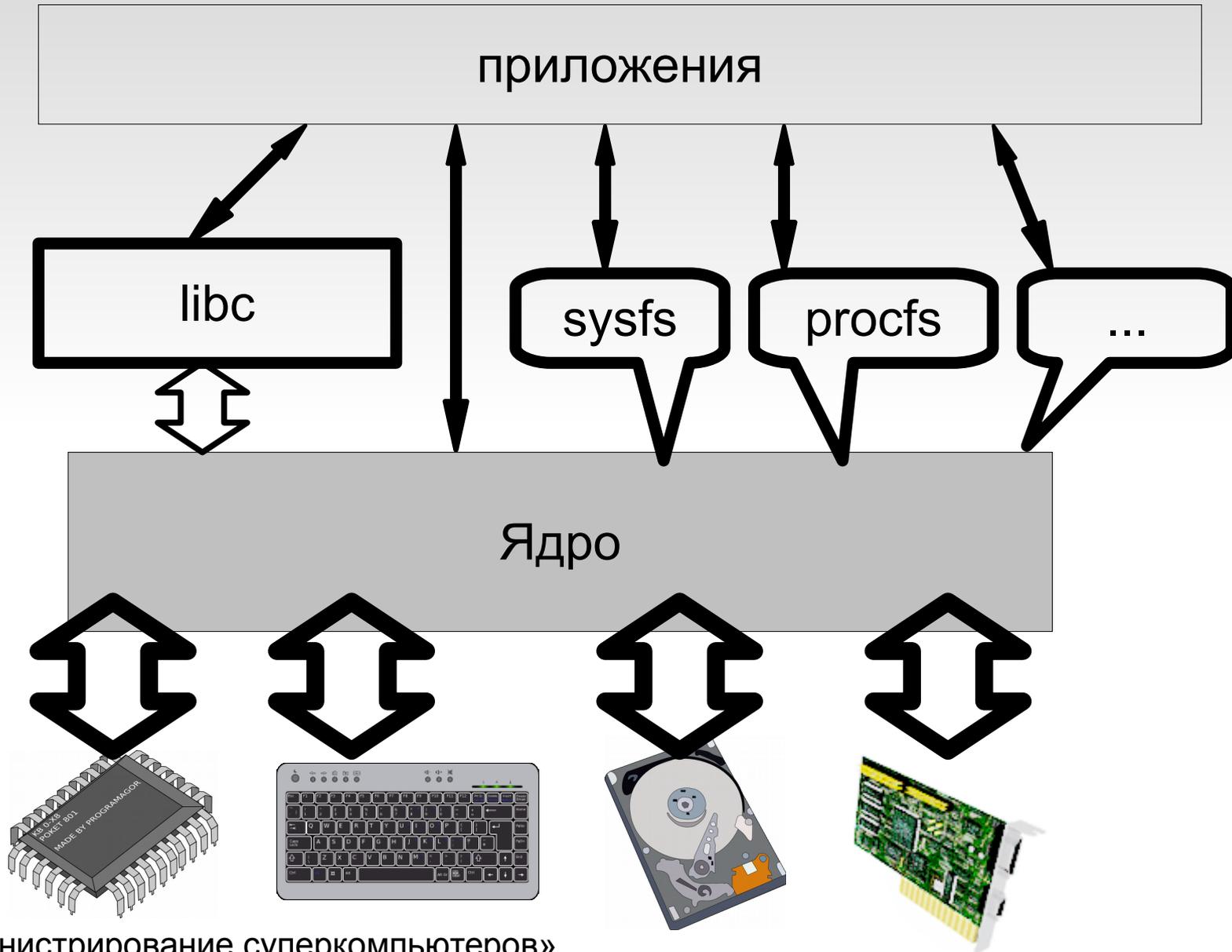


Курс "Администрирование суперкомпьютеров"

Жуматий С.А.

UNIX



UNIX



- lsmod
- modprobe
- insmod
- rmmod
- modinfo

/etc/modules
/etc/modprobe.d/*
/etc/modutils.d/*

UNIX

INIT (sysV)

/etc/rc.d/rcX.d/SNNservice

/etc/init.d/service start/stop

/etc/inittab

Upstart/Systemd/...

/etc/init

service start/stop/restart/reload

UNIX

INITTAB

<id>:<runlevels>:<action>:<process>

id: — любой идентификатор. По традиции — 2 символа

«id» - специальный, указывает runlevel по умолчанию

runlevels — список runlevel-ов, к которому имеет отношение правило

UNIX

INITTAB

<id>:<runlevels>:<action>:<process>

action — действие, одно из:

- initdefault
- sysinit
- boot, bootwait
- wait
- respawn
- once
- power*
- ctrlaltdel
- off
- ... (смотрите в man, но скорее для интереса)

UNIX

INITTAB

примеры

id:2:initdefault:

si::sysinit:/etc/init.d/rcS

~~:S:wait:/sbin/sulogin

l6:6:wait:/etc/init.d/rc 6

1:2345:respawn:/sbin/getty 38400 tty1

ОСНОВЫ Linux

INIT (systemd/upstart/...)

/etc/init.conf

/etc/init/rc-sysinit.conf

start/stop/restart/reload **service**

Основы Linux

SYSTEMD

- нет runlevels, есть цели (targets)
- нет скриптов для сервисов, есть юниты (units)
- юниты бывают:
 - service
 - mount
 - device
 - socket
- * и ещё есть юниты-шаблоны...

ОСНОВЫ Linux

SYSTEMD

`systemctl start/stop/restart/reload` юнит

`systemctl enable/disable/is-enabled` юнит

`systemctl [list-units] [--failed]`

`systemctl daemon-reload`

`systemctl reboot/poweroff/suspend/hibernate`

Основы Linux

SYSTEMD - юниты

`/usr/lib/systemd/system/` - системные

`/etc/systemd/system/` - админские

Запуск:

- по сокету
- по событию D-BUS
- по появлению пути
- по событию udev
- по зависимости

ОСНОВЫ Linux

SYSTEMD - ЮНИТЫ

[Unit]

Description=NFS server and services

Requires=network.target proc-fs-nfsd.mount rpcbind.target

Wants=rpc-statd-notify.service

комментарий

After= network.target proc-fs-nfsd.mount rpcbind.target

Before= rpc-statd-notify.service

Wants=nfs-config.service

After=nfs-config.service

ОСНОВЫ Linux

SYSTEMD - ЮНИТЫ

[Service]

EnvironmentFile=-/run/sysconfig/nfs-utils

Type=oneshot

RemainAfterExit=yes

ExecStartPre=/usr/sbin/exportfs -r

ExecStart=/usr/sbin/rpc.nfsd \$RPCNFSDARGS

ExecStop=/usr/sbin/rpc.nfsd 0

ExecStopPost=/usr/sbin/exportfs -au

ExecStopPost=/usr/sbin/exportfs -f

ExecReload=/usr/sbin/exportfs -r

Оснoвы Linux

udev

```
ls -l /dev/hda
```



```
brw-rw---- 1 root disk 3, 0 ...
```

```
/sys/devices/pci0000\:00/0000\:00\:1f.1/ide0/0.0/uevent
```

```
DRIVER=ide-cdrom
```

```
PHYSDEVBUS=ide
```

```
PHYSDEVDRIVER=ide-cdrom
```

```
MEDIA=cdrom
```

```
DRIVENAME=hda
```

```
MODALIAS=ide:m-cdrom
```

Основы Linux

udev

Правила:

СОВПАДЕНИЕ, [СОВПАДЕНИЕ,] ПРИСВОЕНИЕ [, ПРИСВОЕНИЕ...]

Пример:

```
KERNEL=="loop0", NAME="loop/%n", SYMLINK+="%k"
```

udev

Совпадения: ==	!=
ACTION	имя события
KERNEL	имя устройства в ядре
SUBSYSTEM	имя подсистемы
DRIVER	имя драйвера
TEST{file}	права на файл
PROGRAM	выполнить программу
RESULT	результат выполнения

Основы Linux

udev

Присвоения:

= - присвоить

+= - добавить

:= присвоить и запретить добавления

SYMLINK

создать ссылку

OWNER

сменить владельца (GROUP, MODE)

ATTR

добавить атрибут в sysfs

RUN

запустить программу

Основы Linux

udev

Присвоения:

\$kernel, %k	имя в ядре
\$number, %n	номер в ядре
\$devpath, %p	путь
\$driver	имя драйвера
\$attr{file}, %s	атрибут sysfs
\$env{key}, %E	переменная окружения
\$major, %M	мажорный номер устройства
\$minor, %m	минорный номер устройства
\$result, %c	вывод запуска программы
\$parent, %P	имя родительского устройства
\$name	текущее имя устройства
\$sys, %S	точка монтирования в sysfs
\$\$	\$
%%	%

`udevadm info -a -p /sys/block/sda` - получить информацию по /dev/sda

sysctl

```
net.ipv4.conf.eth1.forwarding = 1
```

```
net.ipv4.conf.default.accept_source_route = 0
```

```
kernel.shmmax = 68719476736
```

ДЕМОНЫ

Syslogd (rsyslog/syslog-ng/klogd)

Sshd

Ntpd

Rpc.* / nfsd / mountd

Named (bind)

Xinetd/inetd

...

UNIX

Файл = inode

Файл существует пока есть на него ссылки

Виртуальные файлы

Устройства — тоже файлы! mknod

UNIX 2

Трюки с shell.

```
# cp file {,.bak}; for i in node-{1,2,3,4}-{1,2,3,4}; ...
```

```
# ls !$
```

```
# sudo !!
```

```
# history | grep sudo
```

```
# !123
```

```
# echo $((123*55+18))
```

UNIX 3

Трюки с shell.

```
# for i in `seq -w 1 100`; do ssh node-$i true;
done
```

```
# abc & xyz & wait
```

```
# abc 3>&1 1>&2 2>&3
```

```
# abc | tee out.log | grep warning
```

\$? = код возврата

\$\$ = pid процесса

UNIX 3

Трюки с shell.

История: Ctrl-R = поиск назад (ещё раз Ctrl-R = искать введённое дальше)

Комментированные команды остаются в истории!

UNIX 4

Полезные команды

grep/egrep/fgrep

[[:digit:]] [[:space:]] [[:alpha:]]

[abcd]

? * + {5,7} | () ← в grep используйте \!

UNIX 4

Полезные команды

find

- -mtime +10 / -mmin
- -type f/d/... -uid/gid -size -perm
- -xdev
- -and -or
- -exec {} ; -delete
- -print -print0

UNIX 5

Полезные команды

```
# xargs -0      -d ';'
```

```
# awk '{print $2}'
```

```
# cut -c 1-20; cut -f 2-3 -d ';'
```

```
# tail -f ...; less
```

```
# pkill -u ...
```

```
# ps auxwww  fax  -eLf
```

PAM

pam: pluggable authentication modules

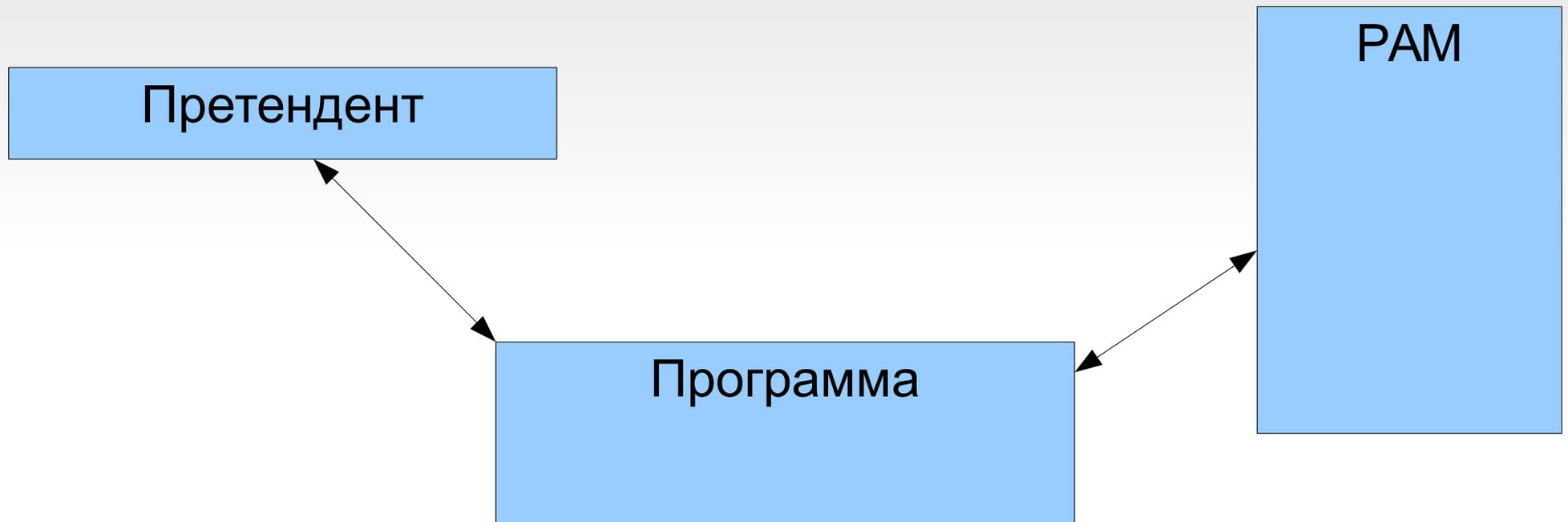
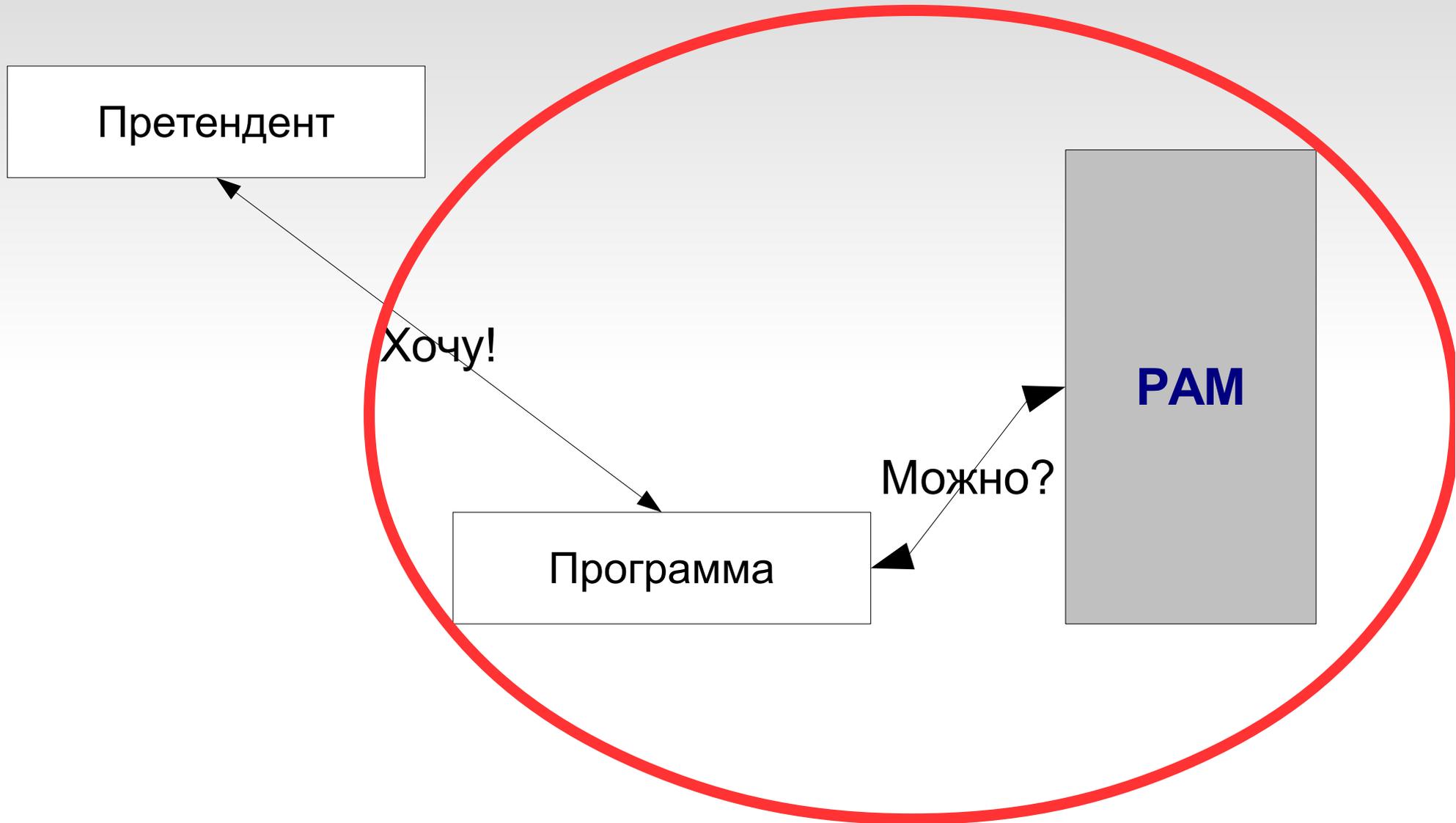


Схема работы



Средства

Средства = возможные запросы клиентских программ

- Auth
- Account
- Session
- Password

Auth

1. Установить «личность»
2. Прописать данные

Account

Проверить доступность аккаунта

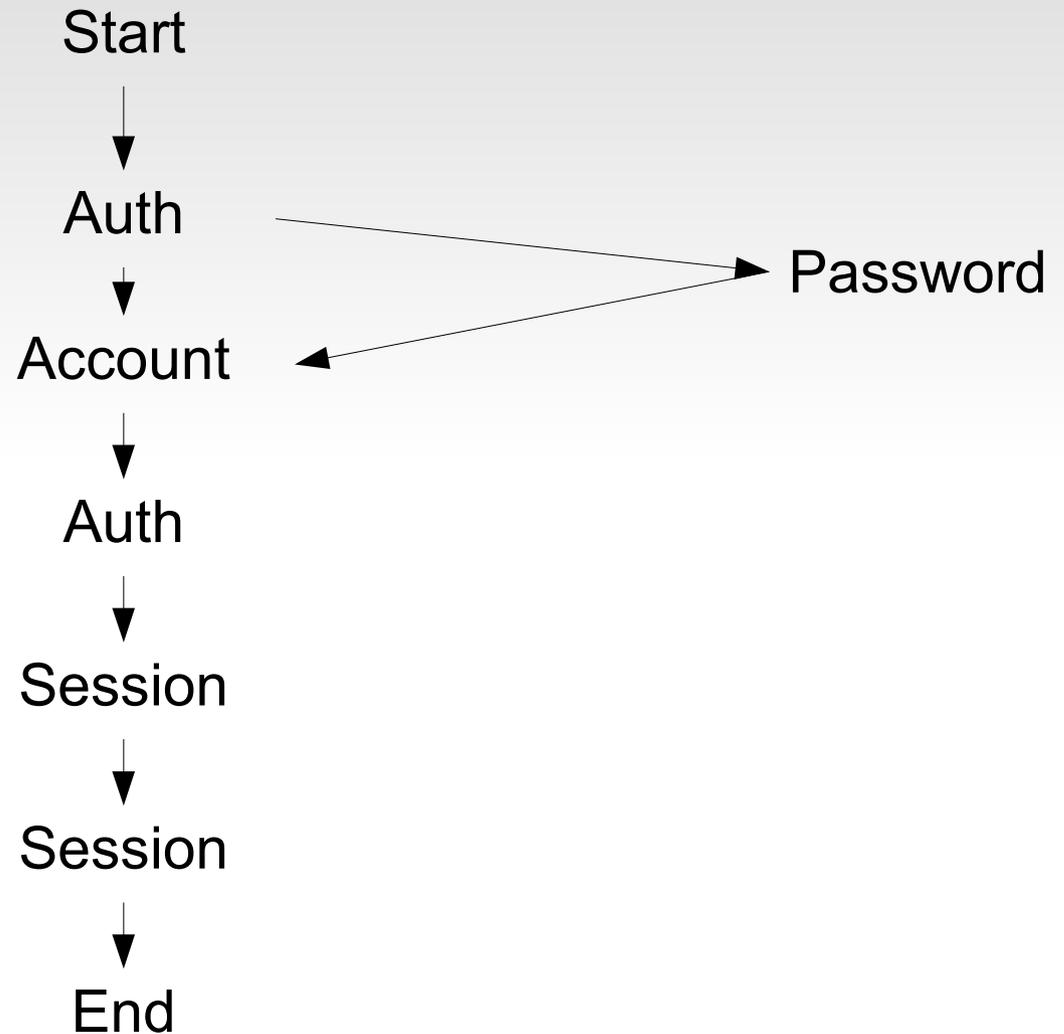
Session

Произвести действия при входе или выходе

Password

Смена пароля

Схема транзакции



Цепочки

Сервис средство флаг модуль [опции]

login auth required nologin.so no_warn
login auth

login account

Флаги

Required:	fail=продолжить цепочку, но запрос отклонить
Requisite:	fail=прервать цепочку и отклонить запрос
Sufficient:	ОК=прервать цепочку и принять запрос*, fail=продолжить цепочку
Optional:	игнорируем результат

Расположение

`/etc/pam.conf`

`/etc/pam.d/service | /etc/pam.d/other`

Полезные модули

`pam_limits`

`pam_exec`

`pam_env`

`pam_listfile`

`pam_warn`

`pam_access`

UNIX

Сеть

```
ifconfig eth0  
ifconfig -a
```

```
ifconfig eth0 1.2.3.4 \  
netmask 255.0.0.0
```

```
ifconfig eth0 down
```

```
route -n
```

```
route add -net 4.5.6.7 \  
netmask 255.0.0.0 \  
gw 4.5.6.254 \  
dev eth0
```

```
route add -net 0.0.0.0 \  
netmask 0.0.0.0 \  
gw 9.8.7.6  
dev eth32
```

UNIX

iproute

ip link show = ifconfig

ip addr show = ifconfig

ip route list = route

ip neigh show = arp

ip rule list = route+