

# CUNC — Console, Unix®, Networks, C (ANSI) Экстремальный минимум для программистов

Есть Проблема :

Программисты, которые приходят к нам на работу  
или на стажировку  
не знают того, что излагается в этом курсе.

**А ДОЛЖНЫ!**

© Евстропов А.В. (Eustrop) & EustroSoft.org

Москва, Санкт-Петербург, Долгопрудный, Зеленоградский  
2019-2020

<http://cunc.eustrosoft.org>



LICENCE: БАЛЭС, BALES, ISC, MIT, BSD на Ваш выбор

# CUNC День 1 (вторник или пятница)

## Ведение в UNIX® - like системы и сети

### 1 === Теория (4 ак. часа)

- 1 25 **О курсе** и предпосылках к его созданию
- 2 45 **Unix** - краткая история, принципы (все есть файл, дерево каталогов, пользователи, процессы, fork/exec, tty, sh, /bin/\*, /etc/\*)
- 3 45 **Console** - shell (sh,csh,ash,bash), переменные окружения (PATH, TERM, EDITOR, PAGER, MM\_CHARSET, LANG), stdin, stdout, stderr, конвейеры (pipes), команды: pwd, ls, cp, mv, cat, man, vi
- 4 45 **Networks** - Модели OSI и DoD, IPv4 - адрес, маска сети, ICMP, UDP, TCP, заголовок пакета IP, Ethernet, ARP, команды: ifconfig, ping, arp, traceroute, telnet, route, netstat -r, nslookup (dig,host) файлы: /etc/hosts, /etc/resolv.conf, /etc/protocols подсматривание трафика через /dev/bpf: tcpdump, tcpflow, trafshow
- 5 30 **C** - список конструкций языка (if, while, ...), базовые типы (char, int, short, long), функция main(..), аргументы командной строки, работа с файлами (open, read, write, close), работа с процессами (fork/exec/kill/exit) компиляция и запуск (cc hello.c -o hello)

### 2 === Практика (4 ак. Часа)

- 1 Почему BSD? (действительно, эту практику можно делать на любом Linux или \*BSD, minix, Solaris [добавить по вкусу преподавателя])
- 2 **Импровизация** по задачам последнего слайда Дня 1
- 3 Цель — приобретение и закрепление навыков установки и настройки среды

# CUNC День 1/ Лекция 0

О курсе и предпосылках к его созданию (25 минут max)

«Все хорошие программисты, которых я знал, были неплохими системными администраторами» /E/

*«Обычно, программистами работают те, кто этому в университете не учился» /E/*

- 1 Проблема уверенной работы в консоли и написания скриптов автоматизации
  - 1 Написание программ для однократных задач, решаемых в одну строчку
  - 2 Проблема с пониманием ТЗ в форме прототипа из sh/awk скриптов
  - 3 Понимание среды, в которой написанная вами программа будет устанавливаться, эксплуатироваться и дорабатываться напильником в процессе
- 2 Проблема понимания устройства и принципов базовой ОС (Unix®, Linux, BSD)
  - 1 Где что и зачем лежит. Как поставить и настроить то, чего не хватает
  - 2 Как работает разграничение доступа
- 3 Понимание устройства Internet, базовых протоколов и границ их применимости
  - 1 Все современные программы работают в сети
  - 2 и взаимодействуют по протоколам, даже сами с собой, внутри себя
- 4 Язык C есть основа для понимания организации памяти процесса (stack/heap), передачи параметров по ссылке (по указателю) и по значению.
  - 1 границ между процессами и ядром. Взаимодействия между процессами
  - 2 Понимание базовых примитивов ОС, таких как файлы, сокет, сигналы, семафоры, мютексы, разделяемая память.
  - 3 Во что на самом деле превращается программа, чтобы ее мог исполнить CPU
- 5 Для решения будем использовать наш опыт курса NETMandala2018

# CUNC День 1/ Лекция 1 (слайд 1/3)

Unix® - краткая история, принципы (45 минут, 20 мин здесь тах)

«(00:00:00 UTC) 1 января 1970 года (четверг)» - «19 января 2038 года 03:14:07 (вторник)»

1 Unix - краткая история. Об этом можно рассказать много и интересно, но это написано в каждой книжке про Unix, а кому лень читать, может послушать здесь: <http://qr.qxyz.ru/?q=100F101> Здесь — кратко и иначе



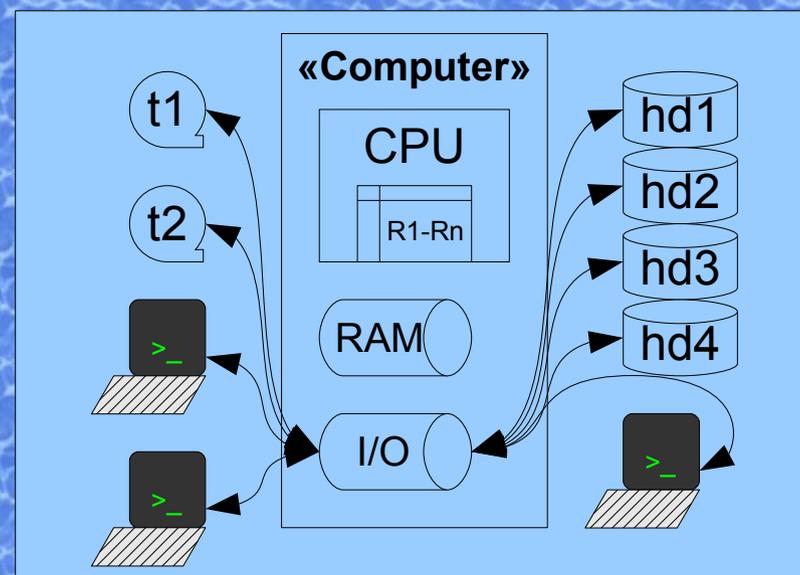
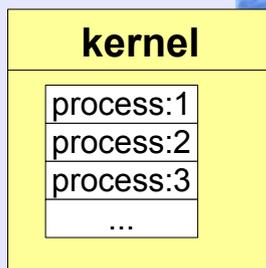
1965	MULTICS	(AT&T/SUN/..)		(DEC)	(DR/MS/IBM)
1969		UNICS			
1970/1971		UNIX V.1, C(1972)		RSX-11	для PDP-11
1975/1974		V6			CP/M
1978/1977			1BSD	VMS	
1979		V7, awk	2BSD	vi, csh	
1980		MS Xenix			QDOS 0.10
1983		X Window	4.2BSD	tcpip GNU	MS-DOS 2.0
1985			Mach		Windows 1.0
1987				MINIX	OS/2
1988		SVR4, SCO		POSIX	
1991		Solaris 2	Net/2	Linux 0.0.1	
1992/1993		UnixWare	386BSD	GNU/Linux	OSF/1 Windows NT 3.1
1994/1995			4.4BSD	Red Hat	Windows 95
			FreeBSD 2, NetBSD 1.1, OpenBSD		
1996					Windows NT 4.0
1999			FreeBSD 3.4	RedHat 6.1	Windows 2000
2000		Снята с эксплуатации последняя инсталляция MULTICS; Beta MacOS X			

# CUNC День 1/ Лекция 1 (слайд 2/3)

Unix® - краткая история, принципы (здесь — 20 мин max)

«(00:00:00 UTC) 1 января 1970 года (четверг)» - «19 января 2038 года 03:14:07 (вторник)»

- 1 Принципы, объяснение от «Computer»
- 2 все есть файл, блочный или символьные
- 3 дерево каталогов, от «/etc/fstab»
- 4 Пользователи, id и login
  - 1 /etc/passwd
  - 2 /etc/group
- 5 Процессы, fork/exec, pid/ppid
- 6 tty — терминал (устройство)
- 7 sh — командный интерпретатор
- 8 /bin/\* - здесь живут команды
- 9 /etc/\* - здесь живут конфиги
- 10 /dev/\* - здесь живут устройства



«/etc/fstab:»

/dev/hd1	/
/dev/hd2	/usr
/dev/hd3	/var
/dev/hd4	swap

(из /dev/hd1)

/bin/sh, csh, ls, cp, mv

/dev/hd1, hd2, t1, tty1

/etc/

/etc/rc

**/etc/fstab**

/root/.profile

/usr/ (из /dev/hd2)

/usr/home/

/usr/home/user1

/usr/home/user2

/var/ (из /dev/hd3)

/var/log/messages

**vmunix**

## «/etc/passwd:»

```
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
operator:*:2:5:System &:/usr/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin
user1:*:101:0>User1:/usr/home/user1:/bin/sh
user2:*:102:102>User2:/usr/home/user2:/bin/csh
```

## «/etc/group:»

```
wheel:*:0:root,user1
operator:*:5:root,user1
user2:*:102:user1
```

## «процессы»

```
1 (init)
2 (/bin/sh /etc/rc)
3 /bin/getty tty1
5 login tty1
10 /bin/sh
11 /bin/ls
6 getty tty2
```

# CUNC День 1/ Лекция 1 (слайд 3/3)

Unix® - краткая история, принципы (5 мин здесь max)  
дополнение — Un\*x сегодня

## 1 Unix - краткая история. Текущее состояние вещей

1965	MULTICS (AT&T/SUN/.)			(DEC)	(DR/MS/IBM)
1969	UNICS				
...	...	...	...	...	...
1988	SVR4, SCO		POSIX		MS-DOS
1991	Solaris 2	Net/2	Linux 0.0.1	OpenVMS	
1992/1993	UnixWare	386BSD	GNU/Linux	OSF/1(?)	Windows NT 3.1
1994/1995		4.4BSD	Red Hat		Windows 95
		FreeBSD 2, NetBSD 1.1			
1996					Windows NT 4.0
1999		FreeBSD 3.4	RedHat 6.1		Windows 2000
2000	Снята с эксплуатации последняя инсталляция MULTICS; Beta MacOS X				
20..	20 лет развития, слияния и поглощения, вытеснение коммерческих UNIX® аналогами на основе BSD/GNU/Linux/WindowsNT, вытеснение CPU архитектурой x86/AMD64 остальных архитектур на периферию (VAX,m68k, SPARC, MIPS, ARM,..)				
2020	Unix® SVR4	BSD	Linux	MS Windows	
	SCO/Unixware, Solaris, IBM AIX;HP-UX; <u>Linux: RadHat Enterprise Linux, CentOS, SUSE</u>	FreeBSD/NetBSD/ OpenBSD/ MINIX, DragonFly, <u>MacOS X</u>	RedHat/Fedora, Debian/Ubuntu,...., <u>Android</u>	Windows XP/7/8/10, <u>Windows Server</u>	

# CUNC День 1/ Лекция 2 (слайд 1/3)

Console - shell, переменные окружения, конвейеры (pipes), команды (sh,csh,ash,bash), (PATH,TERM,EDITOR,PAGER,MM\_CHARSET,LANG), pwd,ls,cp,mv,cat,man,vi

- 1 45 **Console** - shell (sh,csh,ash,bash), переменные окружения (PATH, TERM, EDITOR, PAGER, MM\_CHARSET, LANG), stdin, stdout, stderr, конвейеры (pipes), команды: pwd, ls, cp, mv, cat, man, vi
1. Командный интерпретатор shell (sh,csh,ash,bash), калькулятор bc (bc -l)
2. # man chsh; cat /etc/shells; cat /etc/passwd | grep ^`whoami` | awk -F: '{print \$7}'
3. Команды shell, внутренние и внешние: pwd, cd, ls (ls -al), cp, mv, cat, man
4. # whereis echo; whatis echo; ls /usr/share/man/ | more # or less;
5. # ls /bin ; ls /usr/bin | wc -l; ls /usr/local/bin
6. # touch f1; cp f1 f2; echo hello > f2; cat f2; mv f2 f3; cat f1 f2 f3; ls -al f?
7. # mkdir d1; rmdir d1; mkdir d2; mv f? d12; rmdir d2; cd d2; ls -al
8. # cd .. && rm -r d2
9. # set | more; echo \$PATH # TERM,EDITOR,PAGER,MM\_CHARSET,LANG
10. Так что-же такое «переменные окружения»? (объяснение от fork и exec)
11. Где хранятся переменные окружения?
12. Что такое переменные shell? ; # MYSH\_VAR=value; export MYSH\_VAR
13. Локализация и интернализация; I18N/L10N
14. # man environ
15. # man sh # (ash,bash,ksh,zsh), Особенности csh ( tcsh )

# CUNC День 1/ Лекция 2 (слайд 2/3)

Console - shell, переменные окружения, конвейеры (pipes), команды (sh,csh,ash,bash), (PATH,TERM,EDITOR,PAGER,MM\_CHARSET,LANG), pwd,ls,cp,mv,cat,man,vi

1 stdin, stdout, stderr, конвейеры (pipes); популярные команды grep, awk, xargs

(1) # ls -al /dev/stdin /dev/stdout /dev/stderr

(2) # find /etc > /dev/null ; find /var > /dev/null

(3) # find /etc > /dev/null && find /var > /dev/null

(4) # sh

(1) # find /var > /dev/null 2> /dev/null

(2) # find /var/ | grep -v ^Perm > /dev/null

(3) # find /var/ 2>/dev/stdout | grep -v ^Perm > /dev/null

(5) # csh

(1) # find /var > & /dev/null

(2) # find /var | & grep -v Perm > /dev/null

(3) # find /var | grep -v Perm > /dev/null

(6) # cat /etc/passwd; cat < /etc/passwd

(7) задача преобразования дерева исходных текстов ConceptIS koi8-r → utf-8

(1) # find ./ -type f | xargs file | awk -F : '{ gsub("[ ]", "", \$2); print \$2 ":" \$1 }' | grep -v "^UTF-8Unicodetext" | grep -v "^Javaarchivedata(JAR)" | grep -v "^HTMLdocument,UTF-8Unicodetext" | grep -v "^empty" | grep -v ^data | grep -v ':./hg' | awk -F: '{src=\$2; dst=\$2 ".utf8tmp"; print "cat " \$2 " | iconv -f koi8-r -t utf-8 > " dst "; if [ \"xdiff -q \" dst \" \" src \" = \"x\" ] ; then echo \"no\" ; rm \" dst \" ; else echo \"yes\" ; mv \" dst \" \" src \" ; fi ; }' | sh

# CUNC День 1/ Лекция 2 (слайд 3/3)

Console - shell, переменные окружения, конвейеры (pipes), команды (sh,csh,ash,bash), (PATH,TERM,EDITOR,PAGER,MM\_CHARSET,LANG), pwd,ls,cp,mv,cat,man,vi

Редактор vi — великий и ужасный. от ненависти до любви один шаг

- 1 # vi myfile
- 2 Вопрос: Как выйти из vi? Ответ: нажмите кл. <Esc>, наберите :q! <Enter>
- 3 И если вы хотите использовать другой редактор по умолчанию, пер. окружения:
  - 1 EDITOR=/path/to/favoriteditor; export EDITOR; или setenv EDITOR /path/to/favoriteditor
- 4 У vi три режима: «навигация», «ввод текста» и «командный» (или «ex»)
- 5 Редактирование в режиме «навигация»: dd,yy,p,u, &
- 6 Популярные команды из «навигация» → «ввод текста»: a,i,cw,C,o,O
- 7 Выход из «ввод текста» в «навигация»: <Esc>
- 8 Переход в «командный» из «навигация» - символ «:», обратно - <Enter>
- 9 Популярные команды «командный»: q,q!,w,wq,n,s/это/на это/[g кол-во\_строк]
- 10 # vi hello.c
  - 1 i#include<stdio.h><Esc>ovoid main(){printf("hello!\n");}<Esc>:wq
  - 2 #cc hello.c -o hello
- 11 Всё...

Вопросы?

# CUNC День 1/ Лекция 3 (слайд 1/3)

## Networks - Модели OSI и DoD, Ipv4, Ethernet

(ICMP/UDP/TCP), (ARP), ifconfig, ping, traceroute, nslookup, tcpdump, /etc/host, /etc/resolv.conf

- 1 45 **Networks** - Модели OSI и DoD, IPv4 - адрес, маска сети, ICMP, UDP, TCP, заголовок пакета IP, Ethernet, ARP, команды: ifconfig, ping, arp, traceroute, telnet, route, netstat -r, nslookup (dig,host) файлы: /etc/hosts, /etc/resolv.conf, /etc/protocols  
подсматривание трафика через /dev/bpf: tcpdump, tcpflow, trafshow
- 2 Модели OSI и DoD

OSI	Тип данных	DoD (TCP/IP)	протоколы
7 прикладной		4 прикладной	HTTP,FTP,DNS,telnet,...
6 представления		4	ASCII, EBCDIC, JPG, PNG
5 сеансовый	управление	4	RPC,PAP,L2TP
4 транспортный	датаграммы	3 транспортный	TCP, UDP
3 сетевой	пакеты	2 сетевой	IPv4,IPv6,IPsec,IPX
2 канальный	Кадры (frame)	1 канальный	Ethernet, PPP, ARP
1 физический	биты	1	IEEE 802.3, RS232,USB,E1

# CUNC День 1/ Лекция 3 (слайд 2/3)

## Networks - Модели OSI и DoD, Ipv4, Ethernet

(ICMP/UDP/TCP), (ARP), ifconfig, ping, traceroute, nslookup, tcpdump, /etc/host, /etc/resolv.conf

- 1 **45 Networks** - Модели OSI и DoD, IPv4 - адрес, маска сети, ICMP, UDP, TCP, заголовок пакета IP, Ethernet, ARP, команды: ifconfig, ping, arp, traceroute, telnet, route, netstat -r, nslookup (dig, host) файлы: /etc/hosts, /etc/resolv.conf, /etc/protocols подсматривание трафика через /dev/bpf: tcpdump, tcpflow, trafshow
- 2 IPv4 адрес 32bit: 192.168.25.1 → 11000000.10101000.00011001.00000001
- 3 Ipv4 маска сети: 255.255.255.0 → 11111111.11111111.11111111.00000000
- 4 CIDR запись: # ifconfig ether0 ipv4 192.168.25.1/24
- 5 192.168.25.1/24 и 192.168.25.254/24 — в одной сети
- 6 192.168.25.1/25 и 192.168.25.254/25 — в разных сетях
- 7 В одной сети хосты взаимодействуют напрямую, в разных — через роутер
- 8 # route add 0.0.0.0/0 192.168.25.1; netstat -rn # для windows: # route print
- 9 Для взаимодействия в одной сети им нужно знать MAC-адреса соседа (см. ARP)
- 10 Для обращения по имени — преобразовать в адрес (/etc/hosts, DNS)
- 11 TCP — протокол с установлением соединения и гарантированной доставкой
- 12 UDP — протокол без соединений и не гарантированной доставкой
- 13 ICMP — вспомогательный, для управления внутри других протоколов
- 14 Кроме TCP/UDP/ICMP еще 253 возможных протокола

# CUNC День 1/ Лекция 3 (слайд 3/3)

## Networks - Модели OSI и DoD, Ipv4, Ethernet

(ICMP/UDP/TCP), (ARP), ifconfig, ping, traceroute, nslookup, tcpdump, /etc/host, /etc/resolv.conf

- 1 **45 Networks** - Модели OSI и DoD, IPv4 - адрес, маска сети, ICMP, UDP, TCP, заголовок пакета IP, Ethernet, ARP, команды: ifconfig, ping, arp, traceroute, telnet, route, netstat -r, nslookup (dig, host) файлы: /etc/hosts, /etc/resolv.conf, /etc/protocols подсматривание трафика через /dev/bpf: tcpdump, tcpflow, trafshow
- 2 Иногда, программисту, чтобы разобраться как работает программа, или почему она не работает так, как ожидается, проще и быстрее «подсмотреть» о чем и как она «разговаривает» по сети с другой программой
- 3 Обычно «подсмотреть» проще, чем читать описание сетевого протокола, а даже если прочитать документацию, окажется, что все равно надо смотреть «что-же происходит на самом деле»
- 4 Это не сложно, научимся на практике
- 5 `# tcpdump -pni em0 host 192.168.25.1 and tcp and port 80`

# CUNC День 1/ Лекция 4 (слайд 1/3)

## C - список конструкций языка, базовые типы, файлы

`main(argc,*argv[]), (char/int/short/long), (open/read/write/close); fork/exec/kill/exit; (cc hello.c -o hello)`

- 1 30 C - список конструкций языка (if, while, ...), базовые типы (char, int, short, long), функция main(..), аргументы командной строки, работа с файлами (open, read, write, close), работа с процессами (fork/exec/kill/exit) компиляция и запуск (cc hello.c -o hello)
- 2 Главные книги:
  - 1 Brian W. **Kernighan**, Dennis M. **Ritchie** «The C programming Language»
  - 2 Описание языка программирования ANSI C. (авторов не знаю)
  - 3 Хрестоматия по программированию на Си в Unix, Андрей Богатырев 1998
  - 4 C++ - здесь не рассматриваем
- 3 Синтаксис языка C - основа синтаксиса большинства современных популярных языков программирования (C++, Java, C#, JavaScript, PHP,...). отщепенец Python
- 4 Синтаксис языка C - основа синтаксиса awk, bc, csh. Отщепенцы: sh/ash/bash
- 5 Со знанием языка C можно пройти от океана до океана, и быть понятым
- 6 Почти все конструкции интуитивно знакомы программистам на других языках
- 7 Но — указатели, организация памяти, goto, работа со стеком — добавить надо
- 8 Еще надо понять концепцию препроцессора, ее убрали в Java и C#
- 9 А еще C изначально — препроцессор в assembler, и можно посмотреть во что на самом деле превращается программа для CPU вашего компьютера.

# CUNC День 1/ Лекция 4 (слайд 2/3)

## C - список конструкций языка, базовые типы, файлы

main(argc,\*argv[]), (char/int/short/long), (open/read/write/close); fork/exec/kill/exit; (cc hello.c -o hello)

```
1 #include<unistd.h>
2 #include<stdio.h>
3 #include<stdlib.h>
4 #define STR_SIZE 1024
5 char* NL_SEQ="\n"; // \n for unix, \r\n for DOS/Windows
6 char* sz_ident="CUNC day1.c"; /* comment */
7 char sz_copy[]="(c) EustroSoft.org 2020";
8 #define SL_LIC "LICENCE: BALES || MIT || BSD on your choice"
9 char* sz_lic= SL_LIC ; char sz_buf[STR_SIZE];
10
11 int ccnt_for(char b[]){int i;for(i=0;i<STR_SIZE;i++){if(b[i]==0) break;} return(i);}
12 int ccnt_wh(char b[]){int i=0;while(i<STR_SIZE){if(!b[i]) goto ret; i++;} ret:
    return(i);}
13 int ccnt_goto(char b[]){int i=0;lbl: if(b[i]!=0 && ++i<STR_SIZE){ goto lbl;}return(i);}
14 int (*ccnt)(char*) = ccnt_goto;
15
16 void error(char* errmsg){ fprintf(stderr,"%s%s",errmsg,NL_SEQ); exit(1); } //err&exit
17 void println(str) char* str; {if(str!=0)write(1,str,ccnt(str));write(1,NL_SEQ,2);}
18 /*
19                                     try write(1,NL_SEQ,sizeof(NLSEQ)) */
20 int main()
21 {
22 int i=0; char c='1'; long l=0; float fl=1; double d=3.3;
23 short si = 0; unsigned short usi; char* f=sz_lic,*t=sz_buf;
24 #ifndef __unix__
25 if((99)/(int)(d*10) != 3 ) error("line never printed");
26 if((fl*9.9)/d != 3 ) error("Error: Not Unix-like system detected!");
27 #endif
28 write(1,sz_ident,sizeof(sz_ident)); println(i); println(sz_ident);
29 usi--si; if(usi>0) write(1,sz_copy,sizeof(sz_copy)); println(0);
30 while(i<ccnt_for(f)){t[i]=f[++i];};t[1]=c;write(1,t,ccnt_wh(sz_lic)); println(0);
31 }
```

# CUNC День 1/ Лекция 4 (слайд 3/3)

## C - список конструкций языка, базовые типы, файлы

main(argc,\*argv[]), (char/int/short/long), (open/read/write/close); fork/exec/kill/exit; (cc hello.c -o hello)

```
1 #include<unistd.h>
2 #include<stdio.h>
3 #include <sys/types.h>
4 #include <signal.h>
5 char* sz_ident="CUNC day1_fork.c"; /* comment */
6 char sz_copy[]="(c) EustroSoft.org 2020";
7 char* sz_lic= "LICENSE: BALES || MIT ||BSD on your choice";
8 struct child {pid_t pid; char* s; };
9 #define MAX_CHILDREN 16
10 #define MAX_WAIT 16
11 #define SLEEP_TIME 1
12 struct child ch[MAX_CHILDREN];
13 pid_t my_pid=0; pid_t new_pid=0; int ch_cnt=-1; // child_count
14
15 void sig_handle(int sig) { printf("%s \n",ch[ch_cnt].s); fflush(stdout); }
16
17 int main(int argc,char** argv)
18 {
19 int i=0; char c='1'; long l=0; ch[0].s=sz_ident;ch[1].s=sz_copy;ch[2].s=sz_lic;
20 my_pid=getpid();
21 while(my_pid==getpid())
22 {
23 ch_cnt++; if(ch_cnt>=MAX_CHILDREN) break;
24 if(ch_cnt>3) {ch[ch_cnt].s=sz_ident; if((ch_cnt-3)<argc) ch[ch_cnt].s=argv[ch_cnt-3];}
25 new_pid=fork(); ch[ch_cnt].pid=new_pid;
26 }
27 if(my_pid==getpid()) // I am parent process
28 { while(i<MAX_CHILDREN){sleep(SLEEP_TIME);kill(ch[i].pid,SIGHUP);i++; } }
29 else { // I am child
30     signal(SIGHUP,sig_handle); sig_handle(0); // set signal handler & test it
31     while(i++<MAX_WAIT)sleep(SLEEP_TIME); // wait for MAX_WAIT
32 }
33 } //main()
```

# CUNC День 1/ Перерыв на обед

1

2

3

4

5

А еще, я полагаю, Карфаген должен быть разрушен.

<http://stop-nix.ru>

6

7

8

9

10

11

# CUNC День 1/ Практика (слайд 1/3)

- 1 Почему BSD? (действительно, эту практику можно делать на любом Linux или \*BSD, minix, Solaris [добавить по вкусу преподавателя])
- 2 Установка FreeBSD в виртуальную машину (разбиение дисков, загрузка в singleuser mode, fsck, mount, замена пароля root при потере)
- 3 Добавление и удаление пользователей и групп, vipw (и работа с редактором vi), /etc/passwd, /etc/group и т.п.
- 4 Дерево исходных текстов (где и что лежит)
- 5 Дерево портов и установка пакетов (где, что и как)
- 6 Изменение, перекомпиляция и установка какой-нибудь утилиты, например /usr/src/bin/echo
- 7 Пересборка ядра и мира (buildkernel, buildworld)
- 8 Файлы, hard & soft links, crunch и директория /rescue
- 9 Права доступа к файлу, ugo+rwx, stickybit, chmod, chown, простейший скрипт на sh
- 10 Параметры загрузки /etc/rc.conf, скрипты /etc/rc.d
- 11 Базовая настройка сети, ifconfig, route, netstat, /etc/hosts, /etc/resolv.conf, ping, telnet, traceroute, nslookup (host, dig)

# CUNC День 2 (среда или суббота)

Введение в написание скриптов, RCS/cvs/hg/git, make  
настройка через serial port, VLAN и сети сложнее одноранговых, протоколы, файлы/сокеты

- 1 Введение в написание скриптов (sh, awk, @Hello World! на C), системы управления версиями (rcs, cvs, git). Автоматизация сборки - make (немного про ant и maven). Настройка оборудования через serial port, VLAN и сети - сложнее, чем одноранговые. Важные сетевые протоколы. Работа с файлами и сетями.
- 2 === Теория
  - 1 45 Unix - файлы, устройства (блочные и символьный), сокеты, вызовы ядра.
  - 2 45 Console - Еще раз о конвейерах (pipes); more (less), tail (head), grep, tr, sed, awk, xargs; expr и bc (dc); Регулярные выражения. Инструменты разработчика diff, patch, rcs, cvs, mercurial/git, make (ant, maven) find, sort, uniq, wc; tar, gzip; почему не надо пользоваться cp -r и как надо делать (tar -cf - srcdir/ | tar -pxf - -C /path/to/dstdir/; dos2unix, iconv - проблемы с кодировками, KOI7, KOI-8R, cp866, cp1251, UTF-8 и UTF-16. chroot ^ jail
  - 3 45 Networks - Ethernet - VLAN, 802.1q, Q-in-Q, STP; BOOTP ^ DHCP; чуть-чуть о динамической маршрутизации (BGP, OSPF, RIP); Проблемы протокола TCP, когда он перестает быть адекватным задаче. Классические протоколы первого поколения (telnet, FTP, TFTP, DNS, SMTP, POP) Концепция RPC Протоколы HTTP и SIP/RTP (как устроены и почему я их рассматриваю вместе) Протокол CGI - Великий и Ужасный прародитель современного WEB SSL, SSH, HTTPS, QUIC - настоящее и будущее Internet tcpdump (tcpflow, trafshow, /dev/bpf) - как инструмент, обязательный для разработчика
  - 4 45 C - массивы, указатели строки, организация памяти (stack, heap), переполнение буфера (стека). Демоны, работа от имени root и privsep. Работа с сокетами.
- 3 === Практика

# CUNC День 2/ Лекция 1 (слайд 1/1)

Unix® - файлы, устройства (блочные и символьный), сокеты, вызовы ядра

- 1 # man mkfifo; man dd # разница между блочными и символьными файлами
- 2 /dev/stdin, /dev/stdout, /dev/stderr, /dev/ttyv\*
- 3 /dev/hdd0 и /dev/cd0 — блочные устройства с разным размером блока (512/2048)
- 4 /dev/null, /dev/zero, /dev/random
- 5 /dev/urandom случайные и псевдослучайные чила для криптографии и моделирования методом Монте-Карло. «датчик Франк-Каменецкого»
  - 1 Франк-Каменецкий А.Д. Моделирование траекторий нейтронов при расчете реакторов методом Монте–Карло. – М.: Атомиздат, 1978.
- 6 Локальные сокеты Unix и сокеты TCP/IP (есть суть файлы)
  - 1 # man 2 socket # функции для работы с сокетами в BSD, см SEE ALSO
- 7 # man 2 ioctl; # когда парадигмы «файл» не хватает
- 8 Вызовы ядра, как они устроены и какие бывают (для монолитных ядер)
  - 1 Привилегированный и пользовательский режимы, прерывания
  - 2 Блокирующие и неблокирующие вызовы
- 9 Микроядро — что вместо вызовов ядра (сообщения и т. п.)
  - 1 Проблемы с производительностью и пути их решения
- 10 Виртуализация и «облачные технологии», XEN, VMWare, KVM, VirtualBOX, ...
- 11 SIMH — софтверная эмуляция почти всего (PDP-11,m68k,...), Bochs - ... x86
- 12 Java Virtual Machine (JVM) и JIT, Android и Dalvik

# CUNC День 2/ Лекция 2 (слайд 1/1)

Console - Еще раз о конвейерах, regex, RCS и др., dos2unix, iconv

(more (less), tail (head), grep, tr, sed, awk, xargs; expr и bc (dc); Регулярные выражения.

Инструменты разработчика diff, patch, rcs, cvs, mercurial/git, make (ant, maven) find, sort, uniq, wc; tar, gzip, KOI7, KOI-8R, cp866, cp1251, UTF-8 и UTF-16. chroot ^ jail

- 1 45 Console - Еще раз о конвейерах (pipes); more (less), tail (head), grep, tr, sed, awk, xargs; expr и bc (dc);
- 2 Регулярные выражения.
- 3 Инструменты разработчика diff, patch, rcs, cvs, mercurial/git,
- 4 make (ant, maven)
- 5 find, sort, uniq, wc; tar, gzip;
- 6 почему не надо пользоваться cp -r и как надо делать (tar -cf - srcdir/ | tar -pxf - -C /path/to/dstdir/;
  - 1 Метданные и права доступа ugo+rwX; # chmod 666 myfile; man umask
  - 2 # man ln # hard&soft links
- 7 dos2unix, iconv - проблемы с кодировками
- 8 KOI7, KOI-8R, cp866, cp1251, UTF-8 и UTF-16.
- 9 chroot & jail

# CUNC День 2/ Лекция 3 (слайд 1/3)

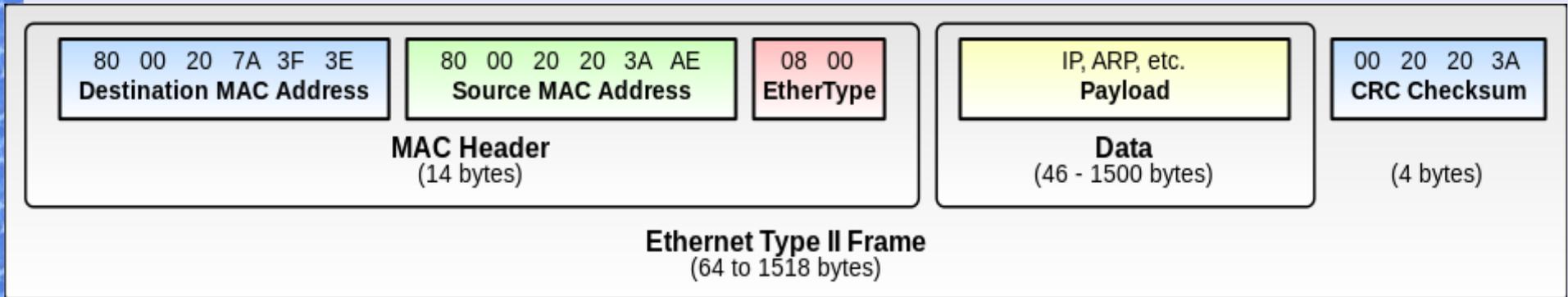
## Networks - Ethernet - VLAN, 802.1q, Q-in-Q, STP; BOOTP ^ DHCP,

- 1 Ethernet - VLAN, 802.1q, Q-in-Q, STP; Формат заголовка IP (отдельный слайд 2шт)
- 2 BOOTP & DHCP;
- 3 чуть-чуть о динамической маршрутизации (BGP, OSPF, RIP);
- 4 Проблемы протокола TCP, когда он перестает быть адекватным задаче.
- 5 Классические протоколы первого поколения (telnet, FTP, TFTP, DNS, SMTP, POP)
- 6 Концепция RPC, мршаллинг/демаршаллинг, проблема byteorder, NBT
- 7 Протоколы HTTP и SIP/RTP (как устроены и почему я их рассматриваю вместе)
- 8 Протокол CGI - Великий и Ужасный прародитель современного WEB
- 9 SSL, SSH, HTTPS, QUIC - настраивающее и будущее Internet
- 10 tcpdump (tcpflow, trafshow, /dev/bpf) - как инструмент, обязательный для разработчика

# CUNC День 2/ Лекция 3 (слайд 1/3)

## Networks - Ethernet и TCP/IP

### 1 Ethernet - VLAN, 802.1q, Q-in-Q, STP; Формат заголовка IP (отдельный слайд)



Отступ	Октет	0								1								2								3							
Октет	Бит	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
0	0	Версия				Размер заголовка				Differentiated Services Code Point				Explicit Congestion Notification				Размер пакета (полный)															
4	32	Идентификатор								Флаги								Смещение фрагмента															
8	64	Время жизни								Протокол								Контрольная сумма заголовка															
12	96	IP-адрес источника																															
16	128	IP-адрес назначения																															
20	160	Опции (если размер заголовка > 5)																															
20 или 24+	160 или 192+	Данные																															



# CUNC День 2/ Лекция 4 (слайд 1/1)

## C - массивы, указатели строки, организация памяти (stack, heap)

переполнение буфера (стека). Демоны, работа от имени root и privsep. Работа с сокетами.

- 1 45 C - массивы, указатели строки, организация памяти (stack, heap), переполнение буфера (стека). Демоны, работа от имени root и privsep. Работа с сокетами.

БК	0010 (pdp-11)	Условно	X86 (32bit)/AMD64
65535	Порты IO	$2^{(\text{sizeof}(\text{int}) * 8) - 1}$	Условно-ядро (наверно)
	Дырка для ПЗУ		Стек (stack)
	Видео-память		Свободная память
	Просто-память		Куча (heap)
	Программа		Программа, сегменты
	Стек		.bss (глобальные переменные, пустые)
			.data (глобальные переменные, не пустые)
			.text (код, константы)
0		0	

# CUNC День 1/ Перерыв на обед

1

2

3

4

5

А еще, я полагаю, Карфаген должен быть разрушен.

<http://stop-nix.ru>

6

7

8

9

10

11

# CUNC День 2/ Практика (слайд 1/3)

- 1 Написание и изучение разнообразных скриптов на sh/awk
- 2 Инструменты разработчика diff, patch, rcs, cvs, mercurial/git, make - в действии
- 3 Установка и настройка WEB-сервера, написание простейшего CGI, применение SSI
- 4 Изучение работы сетевых протоколов посредством tcpdump на примере HTTP,SMTP и POP3
- 5 Разнообразные побочные эффекты и ошибки при программировании на С,
- 6 изучение ассемблерного листинга простейшей программы на С
- 7 Написание "дырявого" сетевого демона на С
- 8 Написание скрипта для его запуска и остановки через /usr/local/etc/rc.d

# CUNC День 3 (четверг или воскресенье)

## Unix как основа Internet. Финальная аттестация.

Internet - как основа всех современных приложений. Взаимопроникновение сетевых технологий и программирования.

- 1 Unix как основа Internet, Internet - как основа всех современных приложений. Взаимопроникновение сетевых технологий и программирования. Финальная аттестация.
- 2 =====Теория
- 3 45 Unix - типовые сервисы NTP, DNS, MAIL (SMTP, POP3), FTP, WEB (Apache, Tomcat, NGINX) Системы мониторинга - NAGIOS и другие Unix как маршрутизатор, firewall, NAT
- 4 45 Console - терминалы, реальные и виртуальные, /etc/termcap, stty, screen (tmux) протокол RS232 (COM-порт), модемы и AT-команды протоколы slip и PPP - туннельные сетевые интерфейсы проброс портов через ssh и прочие приемы
- 5 45 Networks - Как устроен Internet (Краткий обзор-выжимка из курса NETMandala2018) Неочевидные зависимости современных приложений от связанности в Internet. Классы сетей IPv4 (которых больше нет) и классовая маршрутизация Специальные IPv4 адреса по RFC1918 и RFC6598 (и другие) PROXY и NAT - как это работает, проблема-концепция-решение-проблема-... VPN и туннелирование (GRE, IP2IP, PPTP, PPP over UDP, OpenVPN) Концепция LookingGlass, как полигон для P2P-проектов программиста
- 6 45 C - повторение всех перечисленных тем для их закрепления, задачи, которые можно решать на C, которые нужно решать на C, и которые лучше на нем не решать. Синхронизация в многопоточных приложениях (по доступу к файлу, семафоры, мутексы, критические секции)
- 7 === Практика

# CUNC День 3/ Лекция 1 (слайд 1/3)

Unix - типовые сервисы NTP, DNS, MAIL (SMTP, POP3), FTP, WEB

Apache/Tomcat/NGINX, системы мониторинга - NAGIOS и т. п. Unix как маршрутизатор, firewall, NAT

- 1 45 Unix - типовые сервисы
- 2 NTP — сервер точного времени
- 3 DNS — сервер имен, прямая и обратная зона
- 4 MAIL (SMTP, POP3) — любой unix это почтовый сервер ( man mail; ls /var/mail/ )
- 5 FTP — один из старейших протоколов и первый который сломался
- 6 WEB (Apache, Tomcat, NGINX) — Unix как www сервер
- 7 Системы мониторинга - NAGIOS и другие
- 8 Unix как маршрутизатор, firewall, NAT

```
; This is eustrosoft.ru domain
$TTL      3600
@ IN SOA  ns3.glavsv.ru.
root.ns3.glavsv.ru. (
    7      ; Serial
    1200   ; Refresh
    300    ; Retry
    3600000 ; Expire
    1200   ) ; Minimum
IN NS    ns2.glavsv.ru.
IN NS    ns3.glavsv.ru.
IN A     195.208.1.107
IN MX    5  mx02.nicmail.ru.
IN MX    10 mx01.nicmail.ru.
www IN   A      195.208.1.107
www2 IN  A      195.208.1.107
Intra CNAME      www
; L3 DNS zone
nm IN     NS     ns3.glavsv.ru.
```

## CUNC День 3/ Лекция 2 (слайд 1/3)

### Console - терминалы, реальные и виртуальные, RS232

/etc/termcap, stty, screen (tmux) протокол RS232 (COM-порт), модемы и AT-команды протоколы slip и PPP - туннельные сетевые интерфейсы проброс портов через ssh и прочие приемы

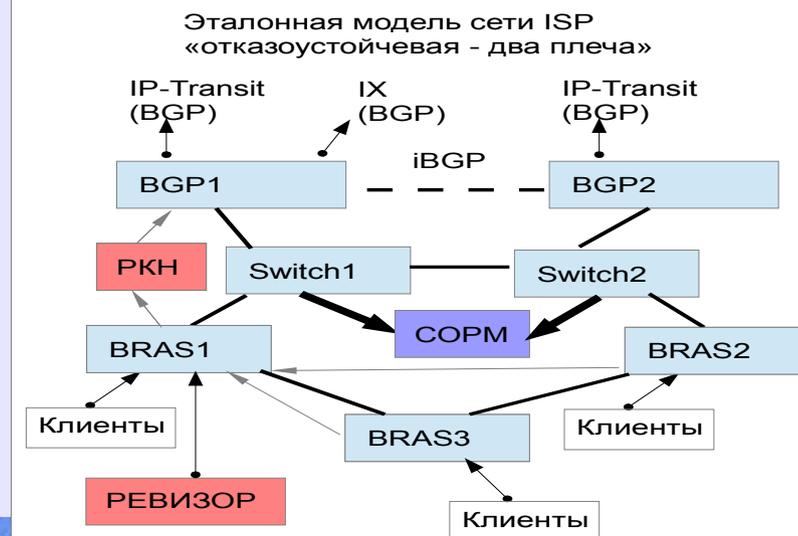
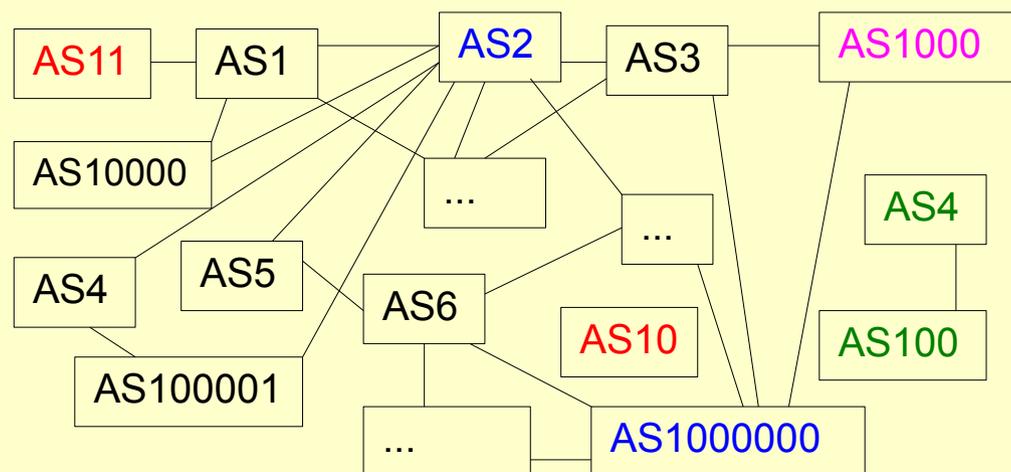
- 1 45 Console - терминалы, реальные и виртуальные, /etc/termcap, stty
- 2 screen (tmux) — виртуальные терминалы на одном терминале
- 3 протокол RS232 (COM-порт)
- 4 модемы и AT-команды
- 5 протоколы slip и PPP - туннельные сетевые интерфейсы
- 6 проброс портов через ssh и прочие приемы

# CUNC День 3/ Лекция 3 (слайд 1)

## Networks - Как устроен Internet

Неочевидные зависимости современных приложений от связанности в Internet, VPN, LookingGlass

- 1 45 Networks - Как устроен Internet (Краткий обзор-выжимка из курса NETMandala2018)
- 2 Неочевидные зависимости современных приложений от связанности в Internet.
- 3 Классы сетей IPv4 (которых больше нет) и классовая маршрутизация
- 4 Специальные IPv4 адреса по RFC1918 и RFC6598 (и другие)
- 5 PROXY и NAT - как это работает, проблема-концепция-решение-проблема-...
- 6 VPN и туннелирование (GRE, IP2IP, PPTP, PPP over UDP, OpenVPN)
- 7 Концепция LookingGlass, как полигон для PЕТ-проектов программиста



# CUNC День 3/ Лекция 4 (слайд 1/3)

## C - повторение всех перечисленных тем

Синхронизация в многопоточных приложениях

- 1 C - повторение всех перечисленных тем для их закрепления,
- 2 задачи, которые можно решать на C, которые нужно решать на C, и которые лучше на нем не решать.
- 3 Синхронизация в многопоточных приложениях
  - 1 по доступу к файлу (man 2 flock; man 2 open)
  - 2 Семафоры (semaphore) - примитив синхронизации работы процессов и потоков, в основе которого лежит счётчик, над которым можно производить две атомарные операции: увеличение и уменьшение значения на единицу, при этом операция уменьшения для нулевого значения счётчика является блокирующей.
  - 3 Мьютексы - (mutex, от mutual exclusion) — аналог одноместного семафора, служащий в программировании для синхронизации одновременно выполняющихся потоков.
  - 4 критические секции (critical section) — объект синхронизации потоков, позволяющий предотвратить одновременное выполнение некоторого набора операций (обычно связанных с доступом к данным) несколькими потоками.
  - 5 Спинлок (Spinlock — циклическая блокировка)
- 4 Разделяемая память (shared memory), memory mapped files

# CUNC День 1/ Перерыв на обед

1  
2  
3  
4  
5

Cetero censeo Carthaginem esse delendam  
<http://stop-nix.ru>

ЭКЗАМЕН!

6  
7  
8  
9  
10

# CUNC День 3/ Практика (слайд 1/3)

- 1 Построение маршрутизируемой сети из компьютеров учебного класса и всего, что окажется под рукой.
- 2 Реализация на каждом узле сети LookingGlass, сделанного на ходу из того, что есть под рукой во FreeBSD.
- 3 Работа с firewall (ipfw) и реализация NAT
- 4 Опционально - реализация LookingGlass демона на C с элементами групповой разработки через mercurial/git, если суммарной квалификации участников для этого окажется достаточно, а также это будет укладываться в отведенное время (что маловероятно).
- 5 ЭКЗАМЕН (20 мин)

## CUNC День 3/ Экзамен, аттестация

### Шкала оценок «по Гамбургскому счету» NETMandata/2018

- 0 — ничего не знает, можно поставить продавником
- 1 — монтажник, 1 ЛП (при наличии соотв. навыков)
- 2 — L2 OSI, 2 ЛП (управление свичами)
- 3 — L3 OSI, сетевой инженер (маршрутизаторы)
- 4 — Главный сетевой инженер, преподавание курса
- 5 — настоящий технический директор, архитектор
- 6 — знает больше, чем этот курс
- 7 — все знает, но overqualified

## CUNC День 3/ Экзамен, аттестация

### Шкала оценок «по Гамбургскому счету» CUNC/2020

- 0 — ничего не знает, можно поставить продавником
- 1 — начинающий пользователь/программист UNIX
- 2 — Уверенный пользователь/программист UNIX
- 3 — Системный администратор UNIX/Devops
- 4 — Главный сисадмин UNIX, преподавание курса
- 5 — настоящий технический директор, архитектор
- 6 — знает больше, чем этот курс
- 7 — все знает, или не все знает, но overqualified