

,

,

. . .

,

UNIX

1...5

«

».

681.3.06

: .,
:

UNIX,
Man, X-Windows,
UNIX,
UNIX,
Shell.
/ ,
, .
UNIX.

9
7 05 2004

.....	1
<i>1</i>	5
1	5
2	5
2.1 UNIX	5
2.2 UNIX	6
2.3 Man	8
2.4 X-Windows	9

3	10
4	10
5	11
	2 _____	11
1	12
2	12
2.1	UNIX	12
2.2	13
2.3	14
2.4	21
2.5	22
3	27
4	28
5	28
	3 _____	29
1	29
2	29
2.1	:	29
2.2	kill	31
2.3	. nice	31
2.4	ps.....	32
2.5	nohup	33
2.6	at	33
2.7	cron	34
3	34
4	34
5	34

	4	35
1	35
2	35
2.1	35
2.2	/ 	37
2.3	noclobber	38
2.4	tee.....	39
2.5	/dev/null	39
2.6	exec.....	40
3	43
4	43
5	43
	5	44
1	44
2	44
2.1	shell.	44
2.2	Shell-	45
2.3	46
2.4	47
2.5	test ("[]").....	47
2.6	"if"	50
2.7	("case").....	51
2.8	("for")	52
2.9	("while")	53
2.10	("until")	54
2.11	55
3	55
4	55
5	55

1

UNIX

1

1.1

UNIX.

UNIX

1.2

UNIX.

1.3

Fr SD.

1.4

X-Windows.

2

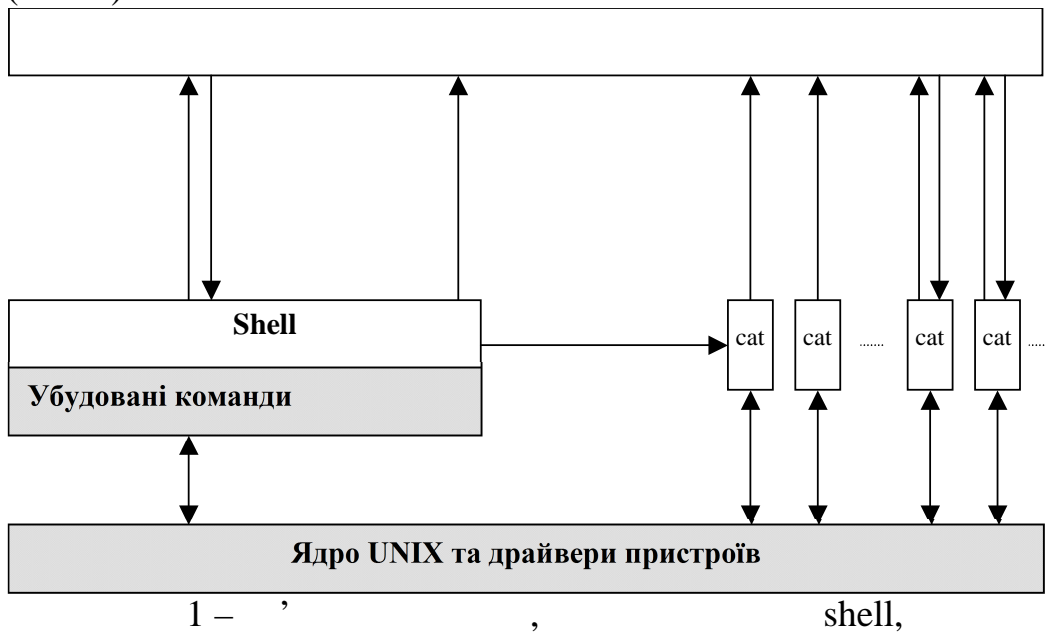
2.1

UNIX

UNIX

Shell.

(. 1).



UNIX “shell” (), shell.

. 1.

sh 1 – Bourn shell () – . B urn shell :

csh shell (). UNIX UNIX- System V. shell Bourn shell, (). C shell.

ksh Korn shell (Bourn shell, shell) (, shell) .

bash “Bourn-again” shell (Free Software Foundation. shell,

tcsh Tcsh – shell. shell, shell, . Bourn shell \$. shell %. (tcsh >).

2.2

UNIX

Cat -відображає зміст зазначеного файлу на термінал (*cat ім'я файлу*);
cd - якщо ви вказали ім'я каталога, то команда **cd** перемінить поточний каталог на зазначений. Якщо ім'я не зазначене, то використовується значення

перемінної оточення \$HOME. Якщо замість імені каталога зазначене ім'я шляху, то ви можете перескочити кілька рівнів за допомогою однієї команди (*cd* ' ');

cd .. – ;

cp - копіює зазначений файл у новий файл, залишивши оригінальний файл незмінним (*cp ім'я джерела ім'я призначення*);

date - відображає поточні дату і час;

diff - порівнює два файли. Команда *diff* видає на стандартне виведення ті рядки файлів, які потрібно змінити, щоб привести файли у відповідність один з одним (*diff файл1 файл2*);

echo - відображає введення на стандартний вивід, включаючи повернення каретки (*echo* " ");

edit - редагування зазначеного файла за допомогою рядкового редактора. Якщо ім'я файла не зазначено, то команда *edit* створює новий файл (*edit* ' ');

grep - пошук по шаблоні, заданому обмеженим регулярним вираженням: *grep* "test" ./* - (*grep* що шукати де шукати);

kill - закінчує фоновий процес за допомогою ідентифікатора процесу (PID) (*kill*);

ls – виводить на термінал імена усіх файлів і каталогів, за винятком тих, котрі починаються з крапки;

mkdir - створює новий каталог. Новий каталог стає підкаталогом того каталога, у якому ви ввели команду *mkdir* (*mkdir ім'я каталога*);

mv - копіювання файла в новий файл у тім же каталозі, або в новий файл в іншому каталозі. Якщо ви переміщуєте файл в інший каталог, то можете використовувати те ж саме ім'я файла (*mv ім'я джерела ім'я призначення*);

more - відображає зміст зазначеного файла на термінал посторінково. Після виводу першої сторінки система робить паузу і чекає від вас підтвердження на продовження виведення наступної сторінки (*more ім'я файла*);

ps - , *ps* PID, *ps* at batch.

pwd - відображає повне ім'я шляху поточного робочого каталога;

rm - видаляє файли чи каталоги з файлової системи. У цій команді ви можете використовувати метасимволи, але з великою обережністю, тому що вилучені файли відновити неможливо (*rm ім'я файла*);

rmdir - видаляє каталог. Ви не можете видалити каталог, у якому знаходитесь, не можете також видалити не порожній каталог (*rmdir ім'я каталога*);

uname - відображає ім'я системи UNIX, у якій ви працюєте;

who - відображає реєстраційні імена користувачів, у даний момент зареєстрованих у вашій системі UNIX;

touch - створення файла (*touch ім'я файла*);

edit – ;

w – (*w ім'я файла*).

2.3

Man.

Man- (Help Microsoft Windows). Man- (man)

UNIX- man- FreeBSD :

1. man- (on-line).

2. man- FreeBSD :

3. man- FreeBSD :

4. man- FreeBSD :

5. man- FreeBSD :

6. man- FreeBSD :

7. man- FreeBSD :

8. man- FreeBSD :

man < >

PAGER. < more > - ,

(1 8),

man < > < man- tty,

man 4 tty - man- man

MANPATH, MANPATH man.

```
setenv MANPATH /home/user/myman:/usr/share/man.
.login, .cshrc, .shrc .profile (
shell )
```

man- /usr/share/man. sh

bash :

```
MANPATH = /home/user/myman:/usr/share/man
export MANPATH
man -k < - > man- ,
< - >.
: man -k mail
aliases(5) - aliases file for sendmail
```


biff(1) - be notified if mail arrives and who it is from
 ctm_smail(1),
 ctm_rmail(1) - send and receive ctm deltas via mail
 forward(5) - mail forwarding instructions
 from(1) - print names of those who have sent mail
 mail(1) - send and receive mail
 mail.local(8) - store mail in a mailbox
 mailaddr(7) - mail addressing description
 mailq(1) - print the mail queue
 mailstats(8) - display mail statistics
 makemap(8) - create database maps for sendmail
 msgs(1) - system messages and junk mail program
 newaliases(1) - rebuild the data base for the mail aliases file
 praliases(8) - display system mail aliases
 rmail(8) - handle remote mail received via uucp
 sendmail(8) - an electronic mail transport agent
 smrsh(8) - restricted shell for sendmail

UNIX

ALT+FX.

(ALT+F1),

2.4 X-Windows

X-Windows -

X- , X-
 ,
 X- X- -
 , (,
 ,). X-
 ,
 X- , X- , display,
 / , X-
 DISPLAY.
 DISPLAY: hostname:displaynumber.screennumber,
 , - :0. :
 setenv hostname:displaynumber.screennumber
 setenv :0

(window managers). X- X- / X- KDE
 GNOME. X- KDE
 - , web- , ,
 , IRC- , , etc.
 ,
 KDE:
 kcontrol - KDE,
 kmail - ,
 konqueror - ,
 kword - ,
 knode - USENET,
 xchat - IRC,
 gimp - .
 X- : /etc/X11/XF86Config.

xf86cfg, xf86config, (X-). Linux ,
 /etc/inittab "id:5:initdefault:". FreeBSD
 X Window display manager.
 X - ,
 /etc/ttys. ,
 startx.
 X- . /.xinitrc.

3

1. ?
2. shell ?
3. .
4. - , - .
5. ?
6. ?

4

1. .
2. .
3. .

5

1. Fr SD.
2. .
3. .
4. X.
 - X- , , xclock.
 - display. X- ,
 - . :
 - DISPLAY. (:0).
 - X- , ,
 - X-window: .
 - xterm. .
 - (window mamager). :
 - /usr/local/bin/startkde&.
5. *xhost*,
 - X- .
 - DISPLAY . X-
 - , , X- .
 - / , ,
6. X- X- X- csh,
 - , : ssh user@host -
 - sh , X-
 - X- DISPLAY .
 - X- ,
 - , X- / ,
7. KDE.
8. , UNIX .
9. , shell ?
10. *edit*.

1.

1.1

UNIX,

1.2

2

2.1

UNIX

UNIX

UNIX

UNIX

, System V Release

4)

UNIX,

UNIX

UNIX (

UNIX.

UNIX

()

(. 2).

UNIX

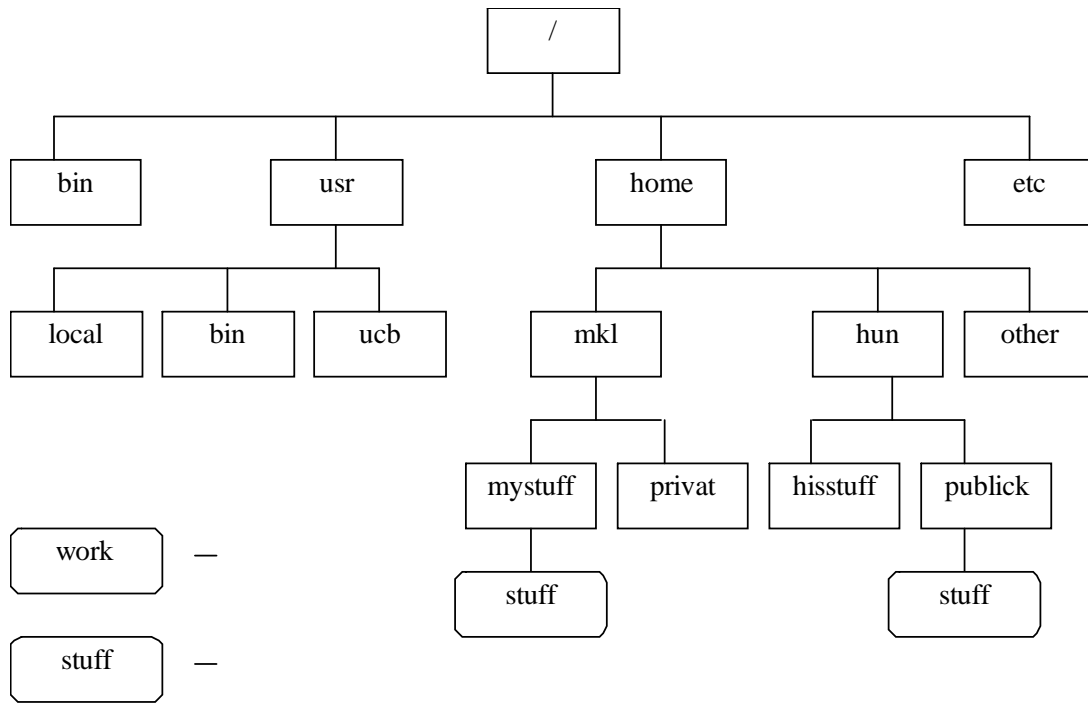
« »

- / (

“ ”,

UNIX «

»



2 -

UNIX

2.2

, UNIX - , (/).

- . , - . , . 2 , . 2 - , ,

- .a
- .c
- .f
- .F
- .gz
- .h
- .html
- .o
- .s
- .z
- .Z
- .1-8

()
 FORTRAN
 FORTRAN
 gzip
 HTML - Web-

. 3 , ,
 . 3 - ,
.txt ASCII-
.tar , *tar*
.shar , shell
.sh Bourne shell
.csh C shell
.mm , *mm*
troff
.ms , *ms* *troff*
.ps PostScript

2.3

- UNIX, FreeBSD - ,
 ,
 . , " " (),
 , ("), " - " "
 , " " "
 - , " " ().
 - , - (, ,
 , . .), " " .
 () (*cron*),
 -" " *inetd*). , (,
 - , -" " -" ").
 , ()
 . - , , , " " ,
 , , , - ,

```

    (passwd)
    ,
    ,
    -
    :
    • (userID).
      us r D
    • , , userID,
      userID
      (cron, inetd . .). " userID -" "
      " " " userID (
      ).
    • userID ,
      , , ,
      " ' " userID - " "
      -
    • userID, "
      userID" " userID" ( " userID"). ( userID
      userID - " userID,
      -" ").
    • ( ) " userID"
      " " " us r D , userID
      us r D , (
      )
    :
      1. ( ) ;
      2. " " ;
      3. . " " (
    ) , , , ,
    , , , ,
    " " ( ) , ,
    . , ,
    . . (userID),
    . « »
    .
  
```



```

    (
    ),
    "w" (write)
    (
    -
    ),
    (
    ),
    (
    "
    "
    "x" (eXecute),
    ("r")
    (
    cd)
    /
    /
    "
    "
    "X".
    "X".
    "
    "
    ("
    "
    )
    "
    "
    ("x"),
    "
    "
    "
    (
    , /usr/share/misc/fonts)
    "w",
    ),
    "
    "
    ("w")
    (
    ).
    "
    "
    permissions.
    "
    "
    root (
    "
    userID"
    ").

```

```
        "x"
,     root
,
        (
,     "
"),
,
        (
)
        (
).

suid.
suid.      Set user ID,
        "
        "
,     "
,     "
        "
userID"
        -
        .
        r t' (
,
        ).
        root.
"     "
        (
,     passwd
        ).
"     "
        -
        ,
        permissions
        **S***** (
        x
        )
        **S***** (
        x
        ).
        suid
        FreeBSD).
        ,
        ,
        ,
        .

sgid.
sgid.      Set group ID,
        "
        .
        ,
        ,
        ,
        .
```

Permissions) *****S** (x) .
 , sgid

sgid
 FreeBSD (BSD-) , , .
 Unix- , , " " .

sticky
 sticky. , " " .
 permissions, *****t (X ")
 ") *****T (X) .

()
 () (w) . (-)
 , , (, , ,
 permissions) .
 (, /tmp) . , " " .

- " " .
 , " " .
 , (, , swap) .
 , ,

--- (, ") :
 r--
 rw- () .
 --- (,) ;
 r-X - ;

```

rwx      (      -w-      --x)      .      ,      .
      .
      -w-      ;
      --x      ,      .
      (      "      "      ).      ,
      ,      permissions      ,
"      "      ".      -wx,      ,      (      -
w-      "      "      (      ,      ).      ,
      .
      r--      ,
      .
      .
      :
      ---      ;
      r-x      "      "      ,
(      /      /      );
      rwx      (      ).
      .
      --x (      ),      -
      r--      ,
      ls,      .
      ,      "      "      /      /permissions
      .      cd.      ,
      (      , r-
x),
      -w-      rw-      ,      -
      --x      -wx      "      "      .      "      "
      ,      -
      ,      -
      .
      "      "      ,      "      -      "
"      '      "      '      ,
"      "      "      "      "
      (      private)      --x.

```

```

private/pictures),
root).
- " "
" " ". -WX
" "
) ..
( ),
:
• - (rwx);
• - (r-x);
• - (---).
root
" " " "
( ) - rwxr-xr-x,
rwx-----.
" "
" _ (
rwx---r-x).
" " "
root)
- " "
" "

```

2.4

```

( " "
- " "
ls ( " -
" - ).
" "
root.
r t'

```

```

"
    root,
    root.

Append -
permission "w").

    "
    append, -
    ( root) -
    (
    ),
    Nounlink - (
    ),
    (
    , permissions
    Immutable change - "
    (
    )
    ,
    (
    )
    ,
    :
    " - uappend, uunlink uimmutable
    " - sappend, sunlink simmutable.
    "
    "
    :
    • archived (
    root,
    "
    "
    ;
    • nodump - "
    "
    ,
    root,
    ,
    nodump
    dump,

```

2.5

, UNIX

, "


```

"          "          suid  sgid).          '          ,          ,          (
          ,          '          ,          '          ).
          ,          ,          .
          -          ,          -          (          )
          .
          ,          root.
          ,          root.
          ,          '          ,          (          )
          chown ("change owner").
          ,
          chgrp ("change group").

          chown
chown
chown <          ><          '          >
chown <          >:<          ><          '          '          >
          ,          "          "          ,
          .
          (          )          "          "          ,          "*"
(          )
          "          "          ,          -R
(recursively).
          ,
chown -R bob:users *
          bob,          users          -
          ,          "          "          ,
          .

          chgrp
chgrp
chgrp <          ><          '          >
          ,          ,          (          )
          ,          ,          chown
          -R,          chgrp.
          (permissions)
          ,          root
(          ).
          ,          ,

```

```

    ,
    .
    ,
    ,
    (permissions)
("change mode").
)
chmod < >< ' >
)
"
"
"
"
u (user)
g (group)
o (other)      a (all)
-
+
=
r
w
x
s      t
      (      ,      "      ").
      ,      ,
      , go
      "
      ,
      "      "      - ugo      a (      ).
      ,      "      "      "      (      -      ).
      ,
      ,
      "      "      "      "      "
      "- "      "+"
      "="
      O+X      -      X      "      ",
      r      w      "      o=rX,
      - "      r      x,      w,
      s      (suid      sgid)
permissions      ,      suid
"      "      u+s,      sgid,      - g+s.
Sticky      chmod a+t ...
      ,
      "      "      ("      ")
      "      "

```

```

chmod a=rx,u+w mydir
r-x, w
) ".
, chmod ( - )
.
, " "
( )
.
, " "
, ( - ).
permissions rwxr-xr-x 755. ,
chmod 755 myfile
" "
chown, chgrp.
-R.
, / - chflags ("change flags").
chflags < > < ' >
< > -
.
sappend - sappnd
uappend - uappnd
sunlink - sunlnk
uunlink - uunlnk
simmutable - schange schg
uimmutable - uchange uchg
arcived - arch
nodump
nosappnd, nosunlnk, noschg . . no -
nodump.
nonodump, dump.
chflags ( -R).

```

3

1. ?
2. ?

- 3. UNIX?
 - 4. , ?
 - 5. , ?
 - 6. ?
 - 7. ?
 - 8. ?
 - 9. , ?
- 4**

- 1. .
 - 2. .
 - 3. .
- 5**

1. , brigada-XX, XX - () . , , .

[brigada-XX]
|
|\
|
|\
|
|\
|

- 2. [brigada-XX].
- 3. 3 .
- 4. , - , . , ,
- 5. , .
- 6. , , 2 , .
- 7. , 1. .
- 8. .

3

1

1.1 FreeBSD.
1.2
1.3

2

2.1

:
 - , ' (' ,
 () , "), (' ,
 ").
 ;
 ;
 ;
 ;
 (VM).
 ,
 , , , ,
 ,
 , , *init*,
 UNIX ,
 : ,
 , UNIX ,
 - , " " *init*
 , UNIX ,
 " " ,
 .

PID.

0 65535.

UNIX

PID.

PID,

- *PPID.*

UID -

(root).

EUID -

UID.

, *EUID* *UID*

suid -

).

, *GID* -

, *EGID* - "

" *GID.*

GID

GID

EGID.

UNIX

login.

(*SUID*) -

SUID-

Shell

PID

init.

swap-

swap-

2.2 kill

kill <- > pid, <- kill.
 > - kill

9 (KILL) - ;
 15 (TERM) - ;
 1 (HUP) - HUP
 HUP

kill.
 " " " "

kill -9 pid,
 - reboot.

2.3 nice

nice- nice.

nice < -“ ”> < >.
 FreeBSD -20 +20.

```

    , : nice,
.
:
# nice +10 /usr/local/mygame

), nice (
, " ".
,
,
FreeBSD renice,
:
renice < > [-p pid] [-g < >] [-u
< >]
:
# renice +1 -p 989 -u daemon root -p 32

1 PID 989 32,
, daemon root.
, ps top.

2.4 ps

ps.

-a - ;
- - ;
-h - PAGER (more less) ;
-m - ;
-r - :
-x - (
, , - );

# ps -ax
(
):
USER PID STAT START TIME COMMAND
bob 1167 R+ 5:57PM 0:00.04 ps -ax

```



```

USER - '
PID -
STAT -
R =
T =
I = (BSD), (System V),
S = (> 20 )
Z =
:
W =
+ =
START -
TIME - '
COMMAND - '
ps.

```

2.5 nohup

```

&
# cat /var/log/messages | grep fetchmail > fetchmail.log &
shell
exit ( Ctrl+D),
(HUP)
).
nohup.

```

```

# nohup cat /var/log/messages | grep fetchmail >
fetchmail.log &

```

```

+5,
nohup.out (
:
csh
nohup.

```

2.6 at

```

at
% at < _
-
[CTRL-d]:

```

% at

1
2

.....

[CTRL-d]

(<)

BSD

UNIX.

at:

at- -

at-s -

shell;
Bourne shell.

at

shell,

atq

atrm

2.7

cron

cron

UUCP-

cron

3

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.

?

?

?

?

?

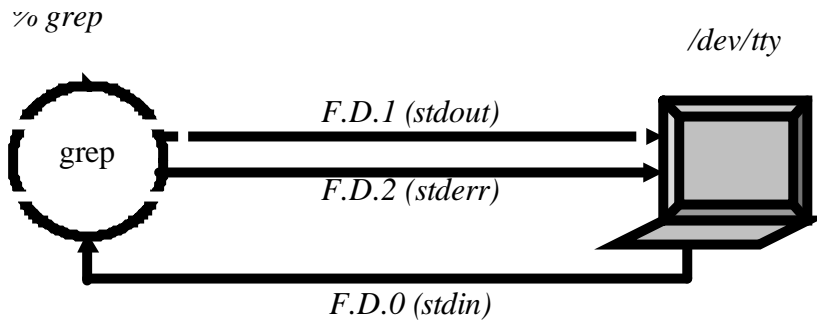
4

- 1.
- 2.
- 3.

5

- 1.
- 2.

- (' , (. 4) (. 3) , (stdin) 0
- (' , (stdout)).
- 1 (' ,).
- (' , (stderr) 2).

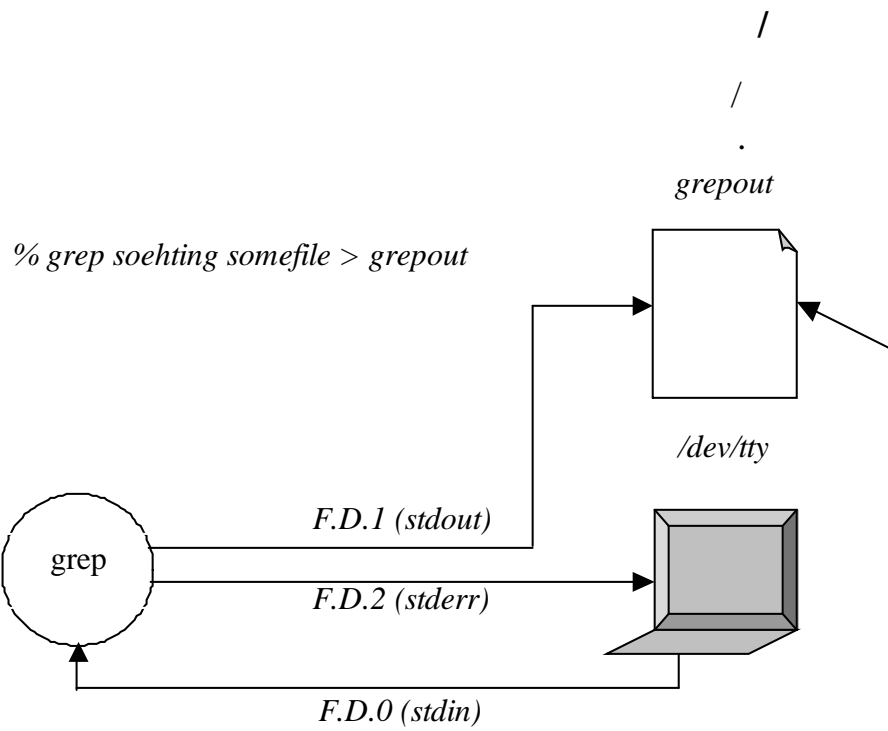


F.D. -

3 -

/

2.2



F.D. -

	4 -	,	/
UNIX,			
	shell (. 4).		
4 -		/	
	csd		sh
stdout	>		>
stderr			2>
stdout stderr	>&		> 2>&1
stdin	<		<
stdout	>>		>>
stderr			2>>
stdout stderr	>>&		>> 2>&1
stdin	>>		>>
stdout		2	
stdout stderr	&	2	

```

1. / :
    < ' _ , joan, shell,
        mail
    :
```

```
% mail joan < myfile
```

```

2. , : - ,
    , :
    % cat file1 file2 > file 3
```

```

( | ( )). UNIX
, joan file1 file2
, :
```

```
% cat file1 file2 | mail joan
```

```

/ .
noclobber
```

```
:
```

```

% > -
, , >> > , ,
.
```

2.3 noclobber

```
noclobber , shell
```

```
,
```

```
shell
```

```
ksh bash. .5
```

```
noclobber.
```

```

5 - noclobber*
   shell tcsh                               ksh bash
% set noclobber                             $ set -o noclobber
% ls                                          $ ls
filea fileb                                 filea fileb
% > fileb                                    $ > fileb
fileb: File exists.                         fileb: fileb: Cannot clobber existing file
% >! fileb                                  $ >| fileb
%

```

2.4 tee

tee

```

. ,
- .
tee.

$ 1 | tee
$ 2 | tee -
$ 3 | tee -
:

$ ( 1
> 2
> 3) | tee

()
- tee.
tee
shell .

```

2.5 /dev/null

/dev/null

```

, UNIX (
), « » - ( ).
:
• ( /dev/null
• );
. ,
% > /dev/null

```

* : *noclobber* , shell
set , *noclobber* ,
shell.

• , , [RETURN].

/dev/null:
% < dev/null.

• , , *grep*

/dev/null, , *grep*

% grep « » * dev/null

2.6 exec

stdin exec

```
#!/bin/bash
#          stdin          'exec'.
exec 6<&0      # '
#6          (stdin).
#          stdin.
exec < data-file # stdin          "data-
file"
read a1      #          "data-
file".
read a2      #          "data-
file."
echo
echo "          ."
echo "-----"
echo $a1
echo $a2
echo; echo; echo
exec 0<&6 6<&-
#          stdin          . #6,
#+          . #6          '          ( 6<&- )
#
```



```

# <&6 6<&-
echo -n "
read b1 # "read",
, stdin.
echo " , stdin."
echo "-----"
echo "b1 = $b1"
echo
exit 0

, exec >filename stdout
. stdout, .

#!/bin/bash
# reassign-stdout.sh

LOGFILE=logfile.txt
exec 6>&1 # ' . #6 stdout.
# stdout.
exec > $LOGFILE # stdout
"logfile.txt".
# - - - - - #
#
$LOGFILE.
echo -n "Logfile: "
date
echo "-----"
echo " \ls -al\"
echo
ls -al
echo; echo
echo " \df\"
echo
df
# - - - - - #
exec 1>&6 6>&- # stdout
. #6.
echo
echo "==" stdout - ==
"
echo
ls -al
echo
exit 0

```

```

                                , stdin stdout,                                exec

#!/bin/bash
# upperconv.sh
#
.
E_FILE_ACCESS=70
E_WRONG_ARGS=71
if [ ! -r "$1" ] # ?
then
    echo " !"
    echo " : $0 input-file output-
file"
    exit $E_FILE_ACCESS
fi # ,
($1) #+ .

if [ -z "$2" ]
then
    echo " ."
    echo " : $0 input-file output-
file"
    exit $E_WRONG_ARGS
fi

exec 4<&0
exec < $1 #
.

exec 7>&1
exec > $2 #
.
# ( ?) .

# - - - - -
# cat - | tr a-z A-Z #
# ^^^^^ # stdin.
# ^^^^^^^^^ # stdout.
# , stdin stdout .
# - - - - -

```

```

exec 1>&7 7>&- # stdout.
exec 0<&4 4<&- # stdin.

#
stdout,
echo " \ "$1\ "
        \ "$2\ ". "

exit 0

3

1. ?
2. ,
   ?
3. ,
4. ?
5. noclobber?

4

1.
2.
3.

5

1. tree.list
2. touch. (: > filename).
3. errors.
4. (stdout) "filename".
5. stdout stderr "filename".
6.
   :
   echo 1234567890 > File #
"File".
   exec 3<> File # "File"
       3.
   read -n 4 <&3 # 4
   echo -n . >&3 #
   exec 3>&- # 3.
   cat File # ==> 1234.67890

7.
   result-file ( cat *.txt | sort | uniq >
result-file).
8. stdin exec.
9. stdout exec.

```

5

shell

1

shell.

2

2.1

shell

shell

date; pwd; ls.

shell

"sh",

"cmd",

\$ sh cmd

\$ sh <cmd

chmod 711 cmd,

cmd

date; pwd; ls

date
pwd
ls

shell.

shell-

```

    .
    shell
"chmod" ,
    UNIX.
    shell ,
    !# '
    .
    #!/bin/sh.
    
```

2.2 Shell-

```

    shell- -
    ,
    ,
    .
    "="
var_1=13 - "13" -
var_2=" UNIX" -
    (" ")
    .
    shell- "="
    ,
    ,
    DAT=`date`
    "date" (
    ),
    ,
    "DAT".
    "read",
    ( )
    "read" "echo",
    - :
    echo -n " : "
    read x
    ,
    :
    , "753", "x".
    "read" ( )
    "read" , (
    
```

```

),
,
,
"read",
,
.
.
shell-
"$".
echo $var_2
echo var_2

```

```

UNIX
var_2

```

2.3

```

. shell
(
) shell-
"$", 0 9:
"examp-1" "tst" "tail".
"1" "2".
( ) ' "0" , "$"
( ):
0 - ;
1 - ;
2 - ..
"many" 13-
many 10 20 30 40 50 60 70 80 90 100 110 120 130
###
# many:
echo "$0: "
echo " = $#
: $1 $5 $9 "
shift
echo "1 : =$1 ' =$5
' =$9"
shift 2
echo "1 + 2 = 3 : =$1 ' =$5
' =$9"
perem=`expr $1 + $2 + $3`
echo $perem

```

`"shift"`
`$1,` `$2, ...`
`!` `$9.`
`:`

many:

`= 13`
`: 10 50 90`
`1 =20 ' =60 ' =100`
`1 + 2 = 3 : =40 ' =80 ' =120`
`150`

2.4

`shell`
`"#".` `,`
`()` `,`
`.`
`#`
`##`
`###`
`shell`

2.5 test ("["]")

`()` `test` `shell.`
`:`
`test`
`[]`
`"test",` `,`
`shell` `"test".` `"["`
`,` `,`
`(, , "expr").`

```

shell                "   ":
•                    (   .6);
•                    (   .7);
•                    (   .8);
•                    .
6 -
-f file              "file"                ;
-d file              "file" -                ;
- file               "file" -                ;
-r file              "file";
-w file              "file";
-s file              "file"                .

(                   "0"),                (
"1"). "specific" - '                .

[ -f specific ] ; echo $?
0
[ -d specific ] ; echo $?
1

7 -
str1 = str2          "str1" "str2"          ;
str1 != str2         "str1" "str2"          ;
-n str1              "str1" (                );
-z str1              "str1" (                ).

x="who is who"; export x; [ "who is who" = "$x" ];
echo $?
0
x=abc ; export x ; [ abc = "$x" ] ; echo $?
0
x=abc ; export x ; [ -n "$x" ] ; echo $?
0
x="" ; export x ; [ -n "$x" ] ; echo $?
1

"test"              "   " (                "0")

[ privet ] ; echo $?
0
[ ] ; echo $?
1

```



```

      ,
      (
      ).

```

```

true ; echo $?
0
false ; echo $?
1

```

8 -

x -eq y	"x"	"y",	
x -ne y	"x"	"y",	
x -gt y	"x"	"y",	
x -ge y	"x"		"y",
x -lt y	"x"	"y",	
x -le y	"x"		"y".

test

" "

, "x",
"x=0".

```

x=abc ; export x ; [ abc -eq "$x" ] ; echo $?
[": integer expression expected before -eq
x=321 ; export x ; [ 321 -eq "$x" ] ; echo $?
0
x=3.21 ; export x ; [ 3.21 -eq "$x" ] ; echo $?
[": integer expression expected before -eq
x=321 ; export x ; [ 123 -lt "$x" ] ; echo $?
0

```

- (not) "
- (or) " "
- (and) " "

```

[ ! privet ] ; echo $?
1
x=privet; export x; [ "$x" -a -f specific ] ;
echo $?
0
x="";export x; [ "$x" -a -f specific ] ; echo $?
1
x="";export x; [ "$x" -a -f specific -o privet ]
; echo $?

```

```

0
x="" ; export x; [ "$x" -a -f specific -o ! privet
] ; echo $?
1

```

2.6 "if"

"if"

```

if
then
    [elif
then
    ]
[else
]
fi

```

"elif" *"else if"* *"if"* (

,). , " "

.

```

[elif
then
]
[else
]

```

, (, - ,

"test"!).

```

if
then
fi

```

, , , .

"fi". *"fi"* *"if"*.

"if-1"

```

if [ $1 -gt $2 ]
    then pwd
    else echo $0 : Hello!
fi

```

```
if-1 12 11
```

```
/home/sae/STUDY/SHELL
```

```
if-1 12 13
```

```
if-1 : Hello!
```

2.7 ("case")

```

"case"
case in
    ) ; ;
    ) ; ;
...
esac
"case" "in" "esac" - " ( "
) " " "
    "esac",
.
.
.
###
# case-1: "case".
# , ,
# "if",
# "case".
echo -n " ? : "
read z
case $z in
    5) echo ! ; ;
    4) echo ! ; ;
    3) echo ! ; ;
    2) echo ! ; ;
    *) echo ! ; ;
esac

```

```

        ";;",
        ";"
    .
        ,
        ";"
    .
        ":",
        "case"
    " - "
        ( $z)
    ")",
    .
    ###
    # case-2:
    #
    #
    case $1 in
        ONE|TWO|THREE) echo : ZERO ;;
        MMM|WWW) echo :
Not-Net ;;
        Hi|Hello|Howdoing) echo : ! ;;
    *) echo ;;
    esac
    .
    ###
    # case-3:
"case"
    echo " ( )
        '
        :
        1 -
        2 -
        3 -
    read x y # x - ' , y -
    case $y in
        1) sort < $x ;;
        2) cat < $x ;;
        3) wc -l < $x ;;
    *) echo " ! " ;;
    esac
2.8 ("for")
        "for" :
    for ' [in ]
        do
    done

```

```

"for" - , "do" "done" -
, "done"! "in"
"

```

```

"lsort"

```

```

for i in f1 f2 f3
do
  proc-sort $i
done

```

```

, "i"
shell- ,
(i=f1, i=f2, i=f3), "procsort".

```

```

###
# print-5:
for i in 1 2 3 4 5
do
  cat file-22 > /dev/lp
done

```

2.9 ("while")

```

"while",
,
,
.

```

```

"while" :
while
do
done

```

```

"while" - , "do" "done")
(
("0")
("break", "continue" "exit").

```

```

###
# print-50: "while"
#
# "file-22"
50

```

```

n=0
while [ $n -lt 50 ] # n < 50
do
    n=`expr $n + 1`
    cat file-22 > /dev/lp
done

```

2.10**(*"until"*)**

```

"until" :
until
do
done

```

"until" -

```

, ( "do" "done" ) ,
("break", "continue" "exit").
"while" ,
( ( ! )
)

```

```

until false
do
    read x
    if [ $x = 5 ]
    then echo enough ; break
    else echo some more
    fi
done

```

```

"enough" "some more"), "5". (
"break" "5".
(" ")

```

```

until date | grep 12:00:
do
    sleep 30
done

```

```

30 . "date"
. "grep"
"date". "12:00:" , "1",
" " , " " 30 ,
) , ( , "grep"

```

2.11

```

:
"while:" . "0". ,
, , "csh".

```

3

1. ? ,
2. , .
3. .
4. , .
5. *test* .
6. *if* .
7. *while* .
8. *for* .
9. *until* .
10. *case* .

4

1. .
2. .
3. .

5

1. test.
2. shell .

- 1. , ' .
- 2. , ' .
- 3. . ' ,
- 4. . LIST , ' .
- 5. ' , ' .
- 6. ' , ' .
- 7. . ' ' .
- 8. , ' .
- 9. , ' ' 345.txt
- 10. , ' ' .