

עבודה ביוניקס וכתובת סקריפטים

מסמך זה הורד מהאתר <http://underwar.livedns.co.il>.
אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע
במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את
מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: underwar@hotmail.com

Home Page: <http://underwar.livedns.co.il>

במסמך זה נציג בקצרה את הפקודות העיקריות איתם נכתוב סקריפטים ביוניקס,
ונציג סיכום של המבנים השונים בהם משתמשים בכתובת סקריפטים. דפים אלו
מיועדים לאנשים המכירים את הנושא הצריכים סיכום מהיר של הנקודות
העיקריות, כדי להיזכר בהן במידת הצורך.
דפים אלו כוללים את עיקרי החומר הדרוש עבור הקורסים "מבוא לתכנות
מערכות" ו"מבוא למערכות תוכנה" בטכניון.

Useful UNIX commands and parameters

Below is short description of useful UNIX commands, with explain about the main parameters.

pwd
print the current dirctory name

cp file1 file2
Copy file1 to file2

cp -r folder1 folder2
copy folder1 and all its sub folders to folder2

ls
show files in current folder

ls -a
show files include hidden files

ls -l
show files with extra details

mkdir name
create new dir with the name `name`

rmdir name
delete dir

chmod mode files
changes file mode
* Comment: you need to set u+x to each script that you write

cat filename
print filename to standart output

more filename
view filename in standart output

mv file1 file2
Move file1 to file2 (or rename file1)

cd folder
change dir to folder

rm file
del file

rm -f file
delete file without ask if you sure

rm -r folder
delete folder and all its sub-folder

head -N file
print the N first lines of the file

tail -N file
print the N last lines of the file

tail +N file
print all without the N last lines of the file

sort file

sort the text in the file (as strings) and print it to the screen
sort -n file
sort the text in the file (as numbers) and print it to the screen
sort -r file
reverse sort
sort -u file
equal lines will be printed only 1 time
grep [options] [^]word [files]
search the `word` in `files`
parameters:
-i - Ignore cases (means 'A' = 'a')
-n - Print also the number of line
-v - Print the lines that `word` don't appear at.
^word - print only lines that begins with `word`
wc -l file
print the number of lines in file
wc -w file
print the number of words in file
wc -c file
print the number of chars in file
echo string
prints the string to the standard output

C-Shell summery

Start of script file:

```
#!/bin/csh  
...
```

If the file is starting with `#!/bin/csh -f`,
there won't be a call to .cshrc

positional parameters:

\$0 - script name

\$n - n`th argument (equals to \$argv[n])

\$#argv - number of arguments (\$0 not included)

\$argv[*] - list of all the arguments (\$0 not included)

Variables

set varname = data

 Create variable with the name varname and put data as its content

In order to get the variable content, we write \$variable

Example:

 echo \$varname

if we want to put arithmetic operation in variable, we use @

example:

 @ blah = 3 + 3

example 2:

 @ blah = \$blah + 1

example 3:

 @ blah ++

foreach

foreach var (wordlist)

 commandlist

end

```
if
----
if (expr) simple-command
```

```
if (expr) then
    commandlist1
[else
    commandlist2]
endif
```

```
switch
-----
switch(str )
    case pattern1:
        commandlist1
        breaksw
    case pattern2:
        commandlist2
        breaksw
    ...
    default:
        commandlist
endsw
```

* pattern - can be a literal string, contains variable substitution, or pattern matching characters (e.g * ? [...])

```
while
-----
while (expr)
    commandlist
end
```

```
break, continue
-----
same as in C
```

csh file query

syntax: -x file

when x can be:

- e - Files exists
- f - Normal file
- x - Executeable file
- d - Directory
- r - Readable
- w - Writeable
- z - Zero Length file

Example:

```
if (-f blahblah) then
    echo file blahblah exist
endif
```

```
if (! -f blah2) then
    echo file blah2 doesn't exist
endif
```

Variable check

Syntax: \$?var or \$(?var)

Example:

```
if(! $?term) then
    set term = vt100
endif
```

stdin

\$< reads line from stdin