**UNX510/DPS918 - Unix BASH Shell Scripting**

**Lecture 13 - Odds and Ends**

**Odds and Ends**

**Some Useful Commands and Options**

* nl - number lines of a file, lots of options for formatting the line numbers:
* ==> cat cars
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> nl cars
* 1 plym fury 77 73 2500
* 2 chevy nova 79 60 3000
* 3 ford mustang 65 45 17000
* 4 volvo gl 78 102 9850
* 5 ford ltd 83 15 10500
* 6 Chevy nova 80 50 3500
* 7 fiat 600 65 115 450
* 8 honda accord 81 30 6000
* 9 ford thundbd 84 10 17000
* 10 toyota tercel 82 180 750
* 11 chevy impala 65 85 1550
* 12 ford bronco 83 25 9525

==> \_

* note that nl results can be duplicated using commands already discussed
  + click here for one possible solution, but try it yourself first
* cat - has some options similar to nl, but with less flexible formatting:
* ==> cat cars.blanklines
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> cat -n cars.blanklines
* 1 plym fury 77 73 2500
* 2 chevy nova 79 60 3000
* 3
* 4
* 5 ford mustang 65 45 17000
* 6 volvo gl 78 102 9850
* 7 ford ltd 83 15 10500
* 8 Chevy nova 80 50 3500
* 9 fiat 600 65 115 450
* 10
* 11
* 12 honda accord 81 30 6000
* 13 ford thundbd 84 10 17000
* 14 toyota tercel 82 180 750
* 15 chevy impala 65 85 1550
* 16 ford bronco 83 25 9525
* ==> cat -b cars.blanklines
* 1 plym fury 77 73 2500
* 2 chevy nova 79 60 3000
* 3 ford mustang 65 45 17000
* 4 volvo gl 78 102 9850
* 5 ford ltd 83 15 10500
* 6 Chevy nova 80 50 3500
* 7 fiat 600 65 115 450
* 8 honda accord 81 30 6000
* 9 ford thundbd 84 10 17000
* 10 toyota tercel 82 180 750
* 11 chevy impala 65 85 1550
* 12 ford bronco 83 25 9525

==> \_

* cat - has an option to display special characters, for example tab displays as ^I, newline displays as $:
* ==> cat cars.tab
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> cat -A cars.tab
* plym^Ifury^I77^I73^I2500$
* chevy^Inova^I79^I60^I3000$
* ford^Imustang^I65^I45^I17000$
* volvo^Igl^I78^I102^I9850$
* ford^Iltd^I83^I15^I10500$
* Chevy^Inova^I80^I50^I3500$
* fiat^I600^I65^I115^I450$
* honda^Iaccord^I81^I30^I6000$
* ford^Ithundbd^I84^I10^I17000$
* toyota^Itercel^I82^I180^I750$
* chevy^Iimpala^I65^I85^I1550$
* ford^Ibronco^I83^I25^I9525$

==> \_

* tac - display lines of a file backwards, last-line first:
* ==> cat cars
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> tac cars
* ford bronco 83 25 9525
* chevy impala 65 85 1550
* toyota tercel 82 180 750
* ford thundbd 84 10 17000
* honda accord 81 30 6000
* fiat 600 65 115 450
* Chevy nova 80 50 3500
* ford ltd 83 15 10500
* volvo gl 78 102 9850
* ford mustang 65 45 17000
* chevy nova 79 60 3000
* plym fury 77 73 2500

==> \_

* tac results can be duplicated using commands already discussed
  + click here for one possible solution, but try it yourself first
* here is an example of displaying the fields in each line backwards, last-field first:
* ==> cat cars
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> awk '{ for (i=NF; i>0; i--) printf("%-8s", $i); printf("\n") }' cars
* 2500 73 77 fury plym
* 3000 60 79 nova chevy
* 17000 45 65 mustang ford
* 9850 102 78 gl volvo
* 10500 15 83 ltd ford
* 3500 50 80 nova Chevy
* 450 115 65 600 fiat
* 6000 30 81 accord honda
* 17000 10 84 thundbd ford
* 750 180 82 tercel toyota
* 1550 85 65 impala chevy
* 9525 25 83 bronco ford

==> \_

* here is an example of a similar awk loop, working with numeric files:
* ==> cat numbers
* 3 5 3 7 9 6 4 2 6
* 1.96 2.95 1.32 5.49 17.54
* 3 1.96 5 2.95 3 1.32 7 5.49 9 17.54 6 4 2 6
* ==> awk '{ s=0; for (i=1; i<=NF; i++) s+=$i; t+=s; print "Subtotal Line " NR ": " s }
* > END { print "Total: " t }' numbers
* Subtotal Line 1: 45
* Subtotal Line 2: 29.26
* Subtotal Line 3: 74.26
* Total: 148.52

==> \_

* bc - a calculator, scale is number of decimal places:
* ==> degrees\_c=21
* ==> echo "scale = 2; $degrees\_c \* 9 / 5 + 32" | bc
* 69.80
* ==> echo "$degrees\_c celcius is $(echo 'scale = 2; '$degrees\_c' \* 9 / 5 + 32' | bc) fahrenheit"
* 21 celcius is 69.80 fahrenheit
* ==> set 80
* ==> echo "scale = 2; ($1 - 32) \* 5 / 9" | bc
* 26.66
* ==> echo "$1 fahrenheit is $(echo 'scale = 2; ('$1' - 32) \* 5 / 9' | bc) celcius"
* 80 fahrenheit is 26.66 celcius

==> \_

* bc results can be duplicated using commands already discussed
  + click here for one possible solution, but try it yourself first
* ls -S - sorts by file size, largest to smallest, -r reverses the order:
* ==> ls -l
* total 24
* -r-------- 1 lczegel users 445 Jun 16 17:59 cars
* -r-------- 1 lczegel users 449 Jun 16 17:59 cars.blanklines
* -r-------- 1 lczegel users 273 Jun 16 17:59 cars.tab
* -rwx------ 1 lczegel users 273 May 27 03:46 trapsig
* -rw------- 1 lczegel users 533 Jun 16 17:59 words
* -rwx------ 1 lczegel users 482 May 27 03:23 yyy
* -rw------- 1 lczegel users 0 May 29 15:58 zzz
* ==> ls -lr
* total 24
* -rw------- 1 lczegel users 0 May 29 15:58 zzz
* -rwx------ 1 lczegel users 482 May 27 03:23 yyy
* -rw------- 1 lczegel users 533 Jun 16 17:59 words
* -rwx------ 1 lczegel users 273 May 27 03:46 trapsig
* -r-------- 1 lczegel users 273 Jun 16 17:59 cars.tab
* -r-------- 1 lczegel users 449 Jun 16 17:59 cars.blanklines
* -r-------- 1 lczegel users 445 Jun 16 17:59 cars
* ==> ls -lS
* total 24
* -rw------- 1 lczegel users 533 Jun 16 17:59 words
* -rwx------ 1 lczegel users 482 May 27 03:23 yyy
* -r-------- 1 lczegel users 449 Jun 16 17:59 cars.blanklines
* -r-------- 1 lczegel users 445 Jun 16 17:59 cars
* -r-------- 1 lczegel users 273 Jun 16 17:59 cars.tab
* -rwx------ 1 lczegel users 273 May 27 03:46 trapsig
* -rw------- 1 lczegel users 0 May 29 15:58 zzz
* ==> ls -Sr
* zzz trapsig cars.tab cars cars.blanklines yyy words

==> \_

* ls -S results can be duplicated using commands already discussed
  + click here for one possible solution, but try it yourself first
* seq - generates sequences of numbers:
* ==> seq 5
* 1
* 2
* 3
* 4
* 5
* ==> seq 1 2 10
* 1
* 3
* 5
* 7
* 9
* ==> seq 1.3 1.5 10
* 1.3
* 2.8
* 4.3
* 5.8
* 7.3
* 8.8
* ==> seq 6 -1.5 -3
* 6.0
* 4.5
* 3.0
* 1.5
* 0.0
* -1.5
* -3.0
* ==> for x in $(seq 1 0.1 2); do echo $x; done
* 1.0
* 1.1
* 1.2
* 1.3
* 1.4
* 1.5
* 1.6
* 1.7
* 1.8
* 1.9
* 2.0

==> \_

* $RANDOM - generates a pseudo-random number between 0 and 32767:
* ==> echo $RANDOM
* 23218
* ==> die1=$((RANDOM % 6 + 1)); die2=$((RANDOM % 6 + 1)); echo "$die1 + $die2 = $((die1 + die2))"
* 2 + 1 = 3
* ==> die1=$((RANDOM % 6 + 1)); die2=$((RANDOM % 6 + 1)); echo "$die1 + $die2 = $((die1 + die2))"
* 5 + 1 = 6
* ==> die1=$((RANDOM % 6 + 1)); die2=$((RANDOM % 6 + 1)); echo "$die1 + $die2 = $((die1 + die2))"
* 6 + 3 = 9
* ==> die1=$((RANDOM % 6 + 1)); die2=$((RANDOM % 6 + 1)); echo "$die1 + $die2 = $((die1 + die2))"
* 5 + 4 = 9
* ==> die1=$((RANDOM % 6 + 1)); die2=$((RANDOM % 6 + 1)); echo "$die1 + $die2 = $((die1 + die2))"
* 3 + 2 = 5

==> \_

* fold and fmt - format text, each with lots of options:
* ==> cat words
* Here are a bunch of words used to demonstrate various
* commands. Some of the lines are quite short and some are quite long. However, they will all be
* used by all the commands. And I've completely run out of things
* to say, so maybe I'll just repeat this paragraph.
* Here are a bunch of words used to demonstrate various
* commands. Some of the lines are quite short and some are quite long. However, they will all be
* used by all the commands. And I've completely run out of things
* to say, so maybe I'll just repeat this paragraph.
* ==> fold -w80 words
* Here are a bunch of words used to demonstrate various
* commands. Some of the lines are quite short and some are quite long. However,
* they will all be
* used by all the commands. And I've completely run out of things
* to say, so maybe I'll just repeat this paragraph.
* Here are a bunch of words used to demonstrate various
* commands. Some of the lines are quite short and some are quite long. However,
* they will all be
* used by all the commands. And I've completely run out of things
* to say, so maybe I'll just repeat this paragraph.
* ==> fmt -w80 words
* Here are a bunch of words used to demonstrate various commands. Some of
* the lines are quite short and some are quite long. However, they will all
* be used by all the commands. And I've completely run out of things to say,
* so maybe I'll just repeat this paragraph.
* Here are a bunch of words used to demonstrate various commands. Some of
* the lines are quite short and some are quite long. However, they will all
* be used by all the commands. And I've completely run out of things to say,
* so maybe I'll just repeat this paragraph.

==> \_

* cut --complement - invert cut field or character selection:
* ==> cat cars
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> tr -s ' ' < cars | cut -d' ' -f3,4
* 77 73
* 79 60
* 65 45
* 78 102
* 83 15
* 80 50
* 65 115
* 81 30
* 84 10
* 82 180
* 65 85
* 83 25
* ==> tr -s ' ' < cars | cut -d' ' -f3,4 --complement
* plym fury 2500
* chevy nova 3000
* ford mustang 17000
* volvo gl 9850
* ford ltd 10500
* Chevy nova 3500
* fiat 600 450
* honda accord 6000
* ford thundbd 17000
* toyota tercel 750
* chevy impala 1550
* ford bronco 9525

==> \_

* here are some examples of using sed to add or replace lines in a file:
* ==> cat cars
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> sed '3 aThis is a new line after the 3rd line' cars
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* This is a new line after the 3rd line
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* ==> sed '/ford/ aThis is a new line after each ford' cars
* plym fury 77 73 2500
* chevy nova 79 60 3000
* ford mustang 65 45 17000
* This is a new line after each ford
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* This is a new line after each ford
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* This is a new line after each ford
* toyota tercel 82 180 750
* chevy impala 65 85 1550
* ford bronco 83 25 9525
* This is a new line after each ford
* ==> sed '/chevy/ cThis is a new line replacing each chevy' cars
* plym fury 77 73 2500
* This is a new line replacing each chevy
* ford mustang 65 45 17000
* volvo gl 78 102 9850
* ford ltd 83 15 10500
* Chevy nova 80 50 3500
* fiat 600 65 115 450
* honda accord 81 30 6000
* ford thundbd 84 10 17000
* toyota tercel 82 180 750
* This is a new line replacing each chevy
* ford bronco 83 25 9525

==> \_