“`echo “ ’ “$f” ’ ” | tr A-Z a-z `“ - vysledok DU A’B

alebo aj takto: “`echo “ \” “$f” \” ” | tr A-Z a-z `“

alebo takto to da shell za mna tie dalsie uvodzovky: “`echo “ ‘$f:q’ ” | tr A-Z a-z `“

ale ono to kvotuje jeden prvok po druhom v skriptoch, cize by to neurobilo to iste

co “ ” kde by to dalo viac prvkov ako jeden prvok

shell berie tie uvodozvky nie tak logicky ako clovek, ale proste prve dalsie co najde tak sparuje s prvymi

priklad 6.csh

napr. set zoznam = ($zoznam[-6]:q sedem} **---** zakomentuje prvych 6 prvkov samostatne z prveho zoznamu a da ich po jednom do noveho zoznamu a este k tomu priradi prvom sedem

priklad 7.csh

echo $cesta:h --- to h odsekne poslednu cast cesty

echo $cesta:h:h --- to h odsekne 2 posledne casti cesty

echo $cesta:t --- to t odsekne poslednu cast doprava, cize to hodi konkretny subor napr. a je jedno, kolko tecok dam kedze to bere posledne z prava

echo $cesta:r --- odsekne priponu a vypise co je pred nou

echo $cesta:e --- vypise len poslednu priponu suboru, ale je dobre dat len posledne e

**ked chcem spustit skrypt a sam si zvolit premenne, tak ho otvorim pomocou cat a potom davam prikazy set a echo**

priklad 9.csh

$0 – nazov aktualneho skriptu

**vsetko je to samostudium z studijnych materialov na OSE su vraj komandy a skripty a to sa od nas ocakava, ze to mame zvladnute na tests**

**kazde slovicko od Brana v zadani je velmi dolezite ze co asi od nas chce ze co vymedzuje**

na teste je lepsie pouzit cez while kopirovanie riadku ako cez foreach, lebo Stainmuler dava velmi dlhe vstupy ktore su vacsie ako 10kB a tak foreach prestane fungovat

**#** v skriptoch znamena pocet pri premennych  
**shift** vezme prvok z pola a posunie ho dolava, cize ho akokeby zabije

**argv** – je vyhradene slovo v skriptoch na premennu a cislovanie je od jednotky, nie od nuly

**exit** - vindem zo skriptu

**breaksw** - je breaknutie switch

**break** – je breaknutie whileu

**endsw** – ukoncuje switch

**sort** **---** sortuje podla prepinaca, bez prepinaca podla abecedy defaultne

**uniq -c ---** spocita vyskyt daneho slova

**csh** – prikaz na debugovanie skriptov

**#! /bin/csh** – toto sa dava na zaciatok skriptov v cshelli, aby sa to vykonavalo za pomoci cshellu

**#** - je komentar

**na konci skriptu musi byt prazdny riadok**

**$# premenna ---** vypise pocet prvkov v poli

**@ cislo = $cislo + 1 ---** inkrementuje premennu

**foreach pom ($premenna:q) --- to :q to kvotuje ako uvodzovky, ale nemozem pouzit “” lebo mi to vypise ze word too long**

**grep -v /patern/ ----** hlada vysledky ktore nematchuju paternu

**greap “ahoj\|homo” ---** ten \| funguje ako OR, cize najde ahoj alebo homo

**cat /subor/ >> /dest/ --- appendne subor na konci dest**

**cat /subor/ > /dest/ --- prepise dest a da tam obsah suboru**

**find /home/OS/predstav/vstupy-2/test3 -type f -printf ‘%T@ %f\n’ ----**

**to %T@ robi to, ze vypise cas poslednej upravy od vzniku UNIX-u, %f\n robi to ze vpise meno suboru a odriadkuje, to printf ‘tu ide vzor’ vypise to co najde podla toho aky je vzor**

vzdy ked spracovavam vstup od uzivatela v skryptoch, tak tam musim dat uvodzovky. to iste plati

aj pri vystupe, aby ak uzivatel zada prikaz, tak aby nevykonalo prikaz ale bralo to ako string

riesenie na testoch budeme musiet vpysovat do suboru

ak sa subor vola **-l** tak **ls ./-l** alebo **ls -- -l**  co znaci, ze za ls uz nie su ziadne prepinace

a s tou logikou s cestou dokazem vela veci vyriesit napr. aj remove **rm**

**Seminár**

**joe /subor/ ---** editor textovy

**vim /subor/ ---** editor textovy

**cd -f /subor/ ~meno ---** prekopiruje subor do mojeho priecinka

**find /home/OS/pocitacove/csh -type f -not \( -name "\*?.?\*" -or -name ".?\*.?\*" \) | wc -l ---------** spocita subory, ktore nemaju pripony, ale treba ratat s tym, ze subor s priponou moze vyzerat aj takto: .nieco.txt preto je tam aj ten druhy patern

**echo -n $MANPATH | wc -c---** kedze echo pridava na koniec vzdycky znak noveho riadka \n, tak ked pouzijem prepinac -n, tak ho odstrani ten znak noveho riadku a teda ak chcem spocitat pocet znakov v premennej mantpath, ktora udava cestu k manuali, tak to zapisem takto

**cut -d ":" -f4 /etc/passwd | grep -c ^`grep "^os2016:" /etc/group | cut -d ":" -f3`\$ /etc/passwd ------** pozor musim osetrovat kedze grep hlada patern v stringu tak mi najde aj substring v stringu cize musim tam davat ten zaciatok a koniec riadku, cize toto mi hlada v etc/group cislo skupiny pri os2016 a potom toto cislo hlada pri studentoch v etc/passwd

**otaznik v cshellovskych paternoch znamena, ze je tem jeden hocijaky znak, \* znamena ze 0 az N znakov je tam**

**find /cesta/ -type f**

**ked chcem dat prikaz do prikazu a tak ze sa to hodi viac ako pajpa v podstate tak tam dam backthick**

**tr /prepinac/ ---** dokaze menit znaky na ine znaky alebo viacery vyskyt scvrknut do jedneho znaku

**tr -c “[:digit:]” “\n” < /subor/ ----** je complement danych znakov a zmeni ich na novy riadok

**tr -cs “[:digit:]” “\n” < /subor/ -----** je complement danych znakov a este ich aj scvrkne na jeden vyskit, preto je tam ten zobak, lebo tr nema parameter subor

**bodkociarka znamena dalsi prikaz v shelli**

**zatvorky su nieco ako spojenie vystupov**

**V JOE**

ked chceme editovat existujuci subor tak musime mat prava na zapis toho daneho suboru

**Ctrl + K** a potom **S ---** ulozil som subor v joe

**CTRL + K, Q ---** zavre joe

**CTRL + K, X ---** ulozi aj zavre joe

**Ctrl + K, H ---**  help

**Na teste**

**cp /src/ /dest/ --- skopiruje a vytvori novy subor s obsahom stareho**

**zadaj ----prikaz na vygenerovanie testu**

**odovzdaj ----prikaz na odtestovanie toho vygenerovaneho testu, preto treba pisat spravne odpovede tam**

**faq si treba pozriet v literature na home/OS/literature\_zakladna/ tu je to faq**

**ale da sa to najst aj cez html stranku**