

Příklady na zápočtovou písemku z úvodu do Unixu, 18. 6. 2004

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře (tj. bez chyb nebo jen s malými chybami) oba příklady.

1.
Napište skript scitmat, který na standardním vstupu dostane matici (soubor, v němž je na každém řádku stejný počet čísel oddělených mezerami). Na výstupu bude matice s přidaným sloupcem, obsahujícím řádkové součty, a s přidaným řádkem, obsahujícím sloupcové součty. V pravém dolním rohu výstupní matice bude součet všech prvků vstupní matice.

2.
Napište skript cpr, který dostane zdrojový a cílový adresář a masku jmen souborů. Skript projde celý strom pod zdrojovým adresářem a soubory se jménem odpovídajícím masce zkopíruje na stejná místa ve stromě pod cílovým adresářem. Vytvoří neexistující adresáře v cílovém adresáři, do nichž potřebuje něco zkopírovat.

=====

Příklady na zápočtovou písemku z úvodu do Unixu, 2. 6. 2004

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře (tj. bez chyb nebo jen s malými chybami) oba příklady.

1.
Napište skript mailgrep, který dostane tři parametry: jméno mailboxu, jméno hlavičky a regulární výraz. Na výstup vypíše maily (ve formátu, jak jsou v mailboxu), které mají v zadané hlavičce hodnotu odpovídající regulárnímu výrazu.

2.
Napište skript, který vypíše 10 nejkratších řádků souboru zadaného jako parametr. Řádky musí zůstat v pořadí, v jakém jsou v původním souboru.

=====

Příklady na zápočtovou písemku z úvodu do Unixu, 20. 5. 2004, skupina 14:00

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře aspoň 50%, tj. dva příklady.

1.
Napište skript, který dostane jako parametr jméno souboru, ve kterém je na každém řádku jméno souboru. Jednotlivá jména jsou postupně nabízena uživateli, který si může u každého souboru vybrat jednu z akcí: X (konec skriptu), L (zobrazit obsah souboru, po zobrazení se opět zeptá na akci pro stejný soubor), R (smazat soubor), N (další soubor).

2.
Napište skript, který pro každého nalogovaného uživatele vypíše jeho login name, domovský adresář a jména terminálů, kde je nalogovaný. Záznam by měl být na jedné řádce ve tvaru: 'login:dir:term1,term2,term3,...' Nalogované

uživatele (včetně jmen terminálů) vypíše příkaz who.

3.

Napište skript, který dostane jako parametr jméno mailboxu a vypíše pro každou adresu odesílatele nalezenou v mailboxu, kolik je v něm mailů od tohoto odesílatele (setříděné podle počtu mailů). Na závěr vypíše celkový počet mailů v mailboxu.

4.

Napište skript, který dostane soubor, v něm je na každém řádku dvojice jméno souboru (bez cesty, soubor existuje v aktuálním adresáři), jméno podadresáře oddělená dvojtečkou. Skript pro každou takovou dvojici vytvoří příslušný podadresář (pokud ještě neexistuje) a v něm symlink, který ukazuje na původní soubor.

=====

Příklady na zápočtovou písemku z úvodu do Unixu, 20. 5. 2004, skupina 9:00

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře aspoň 50%, tj. dva příklady.

1.

Napište skript, který formátuje text po odstavcích (oddělují se prázdným řádkem) do zadaného počtu sloupců. Program vždy vezme odstavec a napíše ho na co nejmenší počet řádků tak, aby žádný řádek nepřesahoval zadanou šířku a k přechodu na nový řádek nedocházelo uprostřed slova.

2.

Napište skript, který pro zadané jméno knihovny (např. c, ssl, apod.) vrátí celé jméno souboru s knihovnou (např. /lib/libc.so.6, /usr/lib/libssl.a). Jméno souboru pro knihovnu XXX má tvar libXXX.a nebo libXXX.so.* a hledá se v /lib, /usr/lib a adresářích zadaných v environmentové proměnné LD_LIBRARY_PATH.

3.

Napište skript pro počítání hodin služeb v laboratoři. Vstupní soubor se skládá z řádků
Příjmení 8:00 12:25
Prázdné řádky a řádky začínající znakem '#' se ignorují. Příkaz vypíše setříděné podle příjmení: příjmení počet hodin (zaokrouhlený na celé hodiny) Kč (40 * počet hodin)

4.

Napište skript wc1c, který se chová stejně jako standardní příkaz wc -l, tj. pro každý soubor na příkazovém řádku vypíše počet řádků v něm a jeho jméno. Pokud bylo souborů více než jeden, vypíše na závěr součet přes všechny soubory. Když není zadán žádný soubor, spočítá řádky standardního vstupu. V tomto případě vypisuje pouze číslo, ne jméno.

=====

Příklady na zápočtovou písemku z úvodu do Unixu, 18. 6. 2004

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře (tj. bez chyb nebo jen s malými chybami) oba příklady.

1.

Napište skript, který najde všechny tagy v HTML souboru, tj. řetězce tvaru

<TAG> nebo </TAG> nebo <TAG attr=val attr2=val> (tag může mít libovolně mnoho atributů). Tag s atributy může být rozdělen na několik řádků. Skript vypíše statistiku jednotlivých tagů, tj. na každém řádku bude jedno jméno tagu (první slovo za otevírací špičatou závorkou) a počet jeho výskytů v souboru. Výstup bude seříděn sestupně podle četnosti.

2.

Napište skript, který bude fungovat jako příkaz tee, tj. obsah standardního vstupu bude průběžně (tak, jak ho čte, řádek po řádku) zapisovat na standardní výstup a do všech souborů zadaných na příkazovém řádku. K řešení nelze použít příkaz tee.

=====

Příklady na zápočtovou písemku z úvodu do Unixu, 26. 5. 2004

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře aspoň 50%, tj. dva příklady.

1. Napište skript, který dostane jako parametr jméno mailboxu a vypíše počet mailů v něm a pro každý mail adresu odesílatele, počet řádků hlaviček a počet řádků těla mailu.

2. Napište skript, který dostane na vstup HTML dokument a na výstup vypíše všechny v něm uvedené odkazy, každý na samostatný řádek. Odkazy jsou v dokumentu ve tvaru , kde mezi A a href může být libovolné množství mezer nebo znaků konce řádku. Místo uvozovek mohou být použity apostrofy. Na jednom řádku může být několik odkazů.

3. Napište skript, který čte standardní vstup a vypíše N nejdelších řádků, kde N je zadáno jako parametr na příkazové řádce. Pokud je skript spuštěn bez parametrů, použije se N=10. Skript musí kontrolovat, že má správný počet parametrů (0 nebo 1).

4. Napište skript, který dostane jako parametry jména adresářů, které prohledá včetně podadresářů a nastaví v nich právo spustitelnosti pro všechny u všech obyčejných souborů (tj. ne adresářů, symlinků, zařízení, apod.) s první řádkou začínající #! nebo pro něž popis typu souboru vypsáný příkazem file obsahuje slovo executable.

=====

Příklady na zápočtovou písemku z úvodu do Unixu, 6. 5. 2004, skupina 14:00

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře aspoň 50%, tj. dva příklady.

1.

Napište skript, který pro každého nalogovaného uživatele (podle výstupu příkazu who) vypíše jeho login name, full name a počet jím vlastněných procesů.

2.

Napište skript newfrom, který dostane jména dvou adresářů DIR1 a DIR2. Soubory z DIR1, které v DIR2 nejsou nebo jsou novější než v DIR2, zkopíruje do DIR2. Skript pracuje rekurzivně i v podadresářích. Pokud přidáme volitelný přepínač -n, skript nebude kopírovat, bude pouze vypisovat, které soubory by se měly zkopírovat.

3.
Napište skript pro editaci souborů, bez použití ed, sed, nebo jiného editoru:
edit -a regexp soubor text - přidá text do souboru za každý řádek obsahující
výskyt zadaného regulárního výrazu
edit -d regexp soubor - smaže každý řádek souboru obsahující výskyt
zadaného regulárního výrazu
Při vkládání znamená ř. 0 "před 1. řádek". Skript musí kontrolovat počet
parametrů.

4.
Napište skript pro obsluhu elektronické nástěnky pomocí mailu. Skript se bude
spouštět z ~/.forward (předpokládá, že dostane vždy jeden mail na standardní
vstup). Když dostane dopis se subjektem "přidej SOUBOR", vytvoří v nějakém
adresáři (může být napevno zadaný ve skriptu) soubor se jménem SOUBOR a uloží
do něj tělo mailu. Při obdržení mailu se subjektem "smaz SOUBOR" smaže zadaný
soubor. Soubory se ukládají pouze do jednoho adresáře, je třeba kontrolovat,
zda jméno souboru v subjectu neobsahuje lomítko. Pokud ano, mail se dál
nezpracovává (ignoruje se).

=====

Příklady na zápočtovou písemku z úvodu do Unixu, 6. 5. 2004, skupina 9:00

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití
jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné
soubory, které vytvoří. Každá instance skriptu by měla používat pomocné
soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně.
Pro úspěšné napsání písemky je třeba mít dobře aspoň 50%, tj. dva příklady.

1.
Napište skript, který pro každého uživatele zadaného na příkazovém řádku
vypíše jména všech skupin, do nichž patří, ve tvaru
uživatel:skupina1 skupina2 skupina3 ...
Do skupiny patří uživatelé, kteří jsou uvedeni v příslušném řádku /etc/group a
uživatelé, kteří mají skupinu nastavenou jako primární v /etc/passwd.

2.
Napište skript whereisman, který dostane jako parametry jméno manuálové
stránky a volitelně i číslo sekce. Na výstup vypíše jména všech adresářů, kde
je manuálová stránka. Stránky hledá v adresářích MAN/man?, kde za MAN postupně
dosazuje adresáře z environmentové proměnné MANPATH a ? je číslo sekce. soubor
s manuálovou stránkou má za jménem ještě příponu s číslem sekce a případně
další příponu, pokud je stránka komprimovaná. např. "whereisman ls 1" vypíše
/usr/share/man/man1.

3.
Napište skript pro editaci souborů, bez použití ed, sed, nebo jiného editoru:
edit -a 10 soubor text - přidá text za řádek 10 do souboru
edit -d 10 soubor - smaže řádek 10 souboru
Při vkládání znamená ř. 0 "před 1. řádek". Skript musí kontrolovat počet
parametrů a to, že zadané číslo řádku není větší než počet řádků.

4.
Napište skript vacation, který půjde spouštět z ~/.forward (předpokládá, že
dostane vždy jeden mail na standardní vstup) a automaticky odpovídá na maily.
Vezme adresu odesílatele a v konfiguračním souboru s řádky tvaru
regexp soubor
hledá 1. regulární výraz, který odpovídá adrese. Pokud je na řádku i jméno
souboru, pošle tento soubor zpět odesílateli původního mailu. Když jméno
souboru chybí, neposílá nic.

=====

Příklady na zápočtovou písemku z úvodu do Unixu

Skripty pište pomocí příkazů probíraných na přednášce/cvičení (tj. bez použití jazyků, jako Perl, Python, Tcl...). Skript po sobě musí rušit veškeré dočasné soubory, které vytvoří. Každá instance skriptu by měla používat pomocné soubory s unikátním jménem, aby mohl být skript spuštěn vícekrát současně. Pro úspěšné napsání písemky je třeba mít dobře (tj. bez chyb nebo jen s malými chybami) oba příklady.

1.

Napište skript, který dostane jako parametr jméno mailboxu a vypíše pro každou adresu odesílatele nalezenou v mailboxu (hlavička From:), jaká je celková velikost mailů (v bajtech) od tohoto odesílatele.

2.

Napište skript, který čte standardní vstup a na standardní výstup z něj zkopíruje pouze ty řádky, jejichž čísla má zadaná jako parametry na příkazovém řádku. Řádky musí být vypsané v pořadí, jak jsou na vstupu, ale parametry mohou být zadané v libovolném pořadí.

=====