

# 1. Úloha (IOS 2016/17)

## Popis úlohy

Cílem úlohy je vytvořit skript, který prozkoumá adresář a vytvoří report o jeho obsahu. Předmětem rekurzivního zkoumání adresáře je počet souborů a adresářů, typy souborů (podle obsahu souboru) a velikosti souborů. Výstupem skriptu je textový report. Skript je konfigurovatelný pomocí příkazové řádky.

## Specifikace chování skriptu

### JMÉNO

`dirgraph` - statistika obsahu adresáře

### POUŽITÍ

`dirgraph [-i FILE_ERE] [-n] [DIR]`

### POPIS

1. Pokud byl skriptu zadán adresář (DIR), bude předmětem zkoumání. Bez zadaného adresáře se prozkoumává aktuální adresář.
2. Přepínač `-i` způsobuje ignorování souborů a adresářů, jejichž název odpovídá rozšířenému regulárnímu výrazu FILE\_ERE. FILE\_ERE nesmí pokrývat název kořenového adresáře.
3. Přepínač `-n` nastavuje normalizaci histogramu (viz níže).

Součástí reportu je:

- informace o adresářích:
- počet všech adresářů,
- počet všech souborů,
- histogram velikosti souborů,
- histogram 10 nejčastějších typů souborů (podle obsahu souboru).

### FORMÁT REPORTU

```
Root directory: DIR
Directories: ND
All files: NF
File size histogram:
    FSHIST
File type histogram:
    FTHIST
```

1. Formát reportu musí přesně (vč. mezer) odpovídat předcházejícímu textu s odpovídajícími čísly.
2. Formát reportu nezahrnuje statistiky ignorovaných souborů či adresářů.
3. DIR je zkoumaný adresář předaný na příkazové řádce při spuštění skriptu. V případě, že skriptu

- nebyl zadán adresář, je DIR aktuální pracovní adresář (cwd).
4. ND (>0) je počet adresářů v adresářovém stromu.
  5. NF (>=0) je počet všech obyčejných souborů.
  6. Víceřádkový histogram velikostí souborů FSHIST je vykreslen pomocí ASCII a je otočený doprava: řádek histogramu udává kategorii a velikost sloupce (resp. řádku vzhledem k otočení histogramu) dané kategorie udává počet souborů v dané kategorii. Každý soubor v dané kategorii je reprezentován jedním znakem mřížka # (v případě normalizovaného histogramu je toto upraveno - viz níže). Každý řádek histogramu začíná dvěma mezerami, názvem kategorie, který je zarovnán na stejnou šířku, následované dvojtečkou a mezerou. FSHISTOGRAM má předem určené kategorie:
    - <100 B
    - <1 KiB
    - <10 KiB
    - <100 KiB
    - <1 MiB
    - <10 MiB
    - <100 MiB
    - <1 GiB
    - >=1 GiB
  7. Histogram typů souborů FTHIST vypadá velmi podobně jako FSHIST. Liší se v kategoriích, které jsou dané typem souborů podle obsahu souboru.
    1. Pořadí jednotlivých kategorií je dáno sestupně podle četnosti souborů v dané kategorii. V případě stejné četnosti není pořadí kategorií definováno.
    2. Kategorie je dána prvními znaky výstupu utility `file` podle stroje, na kterém je utilita spouštěna. Výstup je omezen na 40 znaků. V případě, že byl výstup zkrácen, jsou za název kategorie přidány tři tečky . . . Soubory patří do stejné kategorie, pokud se shoduje jejich takto upravený název typu.
  8. V případě nastavené normalizace histogramu je velikost vykreslené kategorie (tj. počet mřížek) poměrně upravena tak, aby celková délka řádku nepřekročila maximální délku řádku. Maximální délka řádku je dána buď délkou řádku terminálu (pokud je skript spuštěn v terminálu) minus jeden znak, nebo 79, pokud není výstup skriptu terminál.

## NÁVRATOVÁ HODNOTA

Skript vrací úspěch v případě úspěšného zjištění všech informací. Vrací chybu v případě chyby při zjišťování informací o souborech či adresářích. V takovém případě skript skončí také s chybovým hlášením.

## Implementační detaily

- Přítomnost terminálu zjišťujete pomocí utility `test`. Šířku řádku terminálu pomocí `tput cols`.
- Skript by měl mít v celém běhu nastaveno `POSIXLY_CORRECT=yes`.
- Skript by měl běžet na všech běžných shellech (dash, ksh, bash). Ve školním prostředí můžete použít základní (POSIX) `/bin/sh`.
- Referenční stroj neexistuje. Skript musí běžet na běžně dostupných OS GNU/Linux a \*BSD. Ve školním prostředí máte k dispozici počítače v laboratořích (CentOS), stroj merlin (CentOS) a eva (FreeBSD). Pozor, na stroji merlin je shell `/bin/ksh` symbolický odkaz na bash (tj. nechová se jako

Korn shell jako na obvyklých strojích).

- Skript nesmí používat dočasné soubory.

## Příklady použití

Příklady předpokládají skript `dirgraph` v cestě spustitelných programů (PATH).

```
$ dirgraph /usr/local/bin
Root directory: /usr/local/bin
Directories: 1
All files: 85
File size histogram:
<100 B   : #####
<1 KiB   : #####
<10 KiB  : #####
<100 KiB : #####
<1 MiB   : ###
<10 MiB  : #
<100 MiB :
<1 GiB   :
>=1 Gib  :
File type histogram:
ELF 64-bit LSB executable, x86-64, versi...: #####
a /usr/bin/php script executable (binary...: #####
ELF 64-bit LSB shared object, x86-64, ve...: #####
ELF 32-bit LSB shared object, Intel 8038...: #####
a php script, ASCII text executable       : #####
Bourne-Again shell script, ASCII text ex...: ###
C source, ASCII text                       : ###
ELF 32-bit LSB executable, Intel 80386, ...: ###
ASCII text                                 : ##
Bourne-Again shell script, UTF-8 Unicode...: ##

$ dirgraph -n /etc
Root directory: /etc
Directories: 389
All files: 1766
File size histogram:
<100 B   : #####
<1 KiB   : #####
<10 KiB  : #####
<100 KiB : #####
<1 MiB   : ##
<10 MiB  :
<100 MiB :
<1 GiB   :
>=1 Gib  :
File type histogram:
ASCII text                                     : #####
C source, UTF-8 Unicode text                   : ##
PGP public key block Public-Key (old)         : #
XML 1.0 document, UTF-8 Unicode text, wi... : #
UTF-8 Unicode text                           : #
XML 1.0 document, ASCII text                  : #
Python script, ASCII text executable          : #
```

```
POSIX shell script, ASCII text executabl...: #  
XML 1.0 document, UTF-8 Unicode text      : #  
exported SGML document, ASCII text        : #
```