

## 1. Úloha IOS (2021)

### Popis úlohy

Cílem úlohy je vytvořit skript pro analýzu záznamu systému pro obchodování na burze. Skript bude filtrovat záznamy a poskytovat statistiky podle zadání uživatele.

### Zjednodušený úvod do problematiky

Na burze se obchoduje s cennými papíry (např. akcie společnosti, dluhopisy), komoditami (např. ropa, zeli) apod. Každý obchodovaný artikl má jednoznačný identifikátor, tzv. *ticker* (např. akcie firmy Intel mají na burze NASDAQ ticker *INTC*, bitcoin může mít přiřazený ticker *BTC*). Cena artiklů se mění v čase. Obchodník na burze vstupuje do *pozic*, buď tak, že koupí artikl a očekává, že jeho cena poroste, aby jej pak prodal za vyšší cenu (tzv. *dlouhá* pozice), nebo že artikl prodá a očekává, že jeho cena klesne, aby jej poté mohl koupit levněji (tzv. *krátká* pozice). Obchodník může prodát i artikl, který právě nevlastní (v reálu to funguje tak, že si ho od někoho, kdo ho vlastní, "vypůjčí"), prodá jej, a potom ho koupí za nižší cenu a "vrátí". V našem případě budeme uvažovat, že obchodníkův systém posílá na burzu příkazy k *nákupu* (*buy*) nebo *prodeji* (*sell*) určitého množství jednotek artiklu označeného nějakým tickerem.

### Specifikace rozhraní skriptu

#### JMÉNO

- `tradelog` - analyzátor logů z obchodování na burze

#### POUŽITÍ

- `tradelog [-h|--help] [FILTER] [PŘÍKAZ] [LOG [LOG2 [...]]]`

#### VOLBY

- `PŘÍKAZ` může být jeden z:
  - `list-tick` – výpis seznamu vyskytujících se burzovních symbolů, tzv. "tickerů".
  - `profit` – výpis celkového zisku z uzavřených pozic.
  - `pos` – výpis hodnot aktuálně držených pozic seřazených sestupně dle hodnoty.
  - `last-price` – výpis poslední známé ceny pro každý ticker.
  - `hist-ord` – výpis histogramu počtu transakcí dle tickeru.
  - `graph-pos` – výpis grafu hodnot držených pozic dle tickeru.
- `FILTER` může být kombinace následujících:
  - `-a DATETIME` → after: jsou uvažovány pouze záznamy PO tomto datu (bez tohoto data). `DATETIME` je formátu `YYYY-MM-DD HH:MM:SS`.
  - `-b DATETIME` → before: jsou uvažovány pouze záznamy PŘED tímto datem (bez tohoto data).
  - `-t TICKER` – jsou uvažovány pouze záznamy odpovídající danému tickeru. Při více výskytech přepínače se bere množina všech uvedených tickerů.
  - `-w WIDTH` – u výpisu grafů nastavuje jejich šířku, tedy délku nejdelšího řádku na `WIDTH`. Tedy, `WIDTH` musí být kladné celé číslo. Více výskytů přepínače je chybné spuštění.
- `-h` a `--help` vypíšíu nápovědu s krátkým popisem každého příkazu a přepínače.

### Popis

- Skript filtruje záznamy z nástroje pro obchodování na burze. Pokud je skriptu zadán také příkaz, nad filtrovanými záznamy daný příkaz provede.
- Pokud skript nedostane ani filtr ani příkaz, opisuje záznamy na standardní výstup.
- Skript umí zpracovat i záznamy komprimované pomocí nástroje gzip (v případě, že název souboru končí `.gz`).
- V případě, že skript na příkazové řádce nedostane soubory se záznamy (`LOG`, `LOG2` ...), očekává záznamy na standardním vstupu.
- Pokud má skript vypsat seznam, každá položka je vypsána na jeden řádek a pouze jednou. Není-li uvedeno jinak, je pořadí řádků dáno abecedně dle tickerů. Položky se nesmí opakovat.
- Grafy jsou vykresleny pomocí ASCII a jsou otočené doprava. Každý řádek histogramu udává ticker. Kladná hodnota či četnost jsou vyobrazeny posoupností znaku mířky `#`, záporná hodnota (u `graph-pos`) je vyobrazena posoupností znaku vykřičníku `!`.

### Podrobné požadavky

- Skript analyzuje záznamy (logy) pouze ze zadaných souborů v daném pořadí.
- Formát logu je CSV kde oddělovačem je znak středníku `;`. Formát je řádkový, každý řádek odpovídá záznamu o jedné transakci ve tvaru

```
DATUM A CAS;TICKER;TYP  TRANSAKCE;JEDNOTKOVÁ CENA;MENA;OBJEM;ID
```

kde

- `DATUM A CAS` jsou ve formátu `YYYY-MM-DD HH:MM:SS`
- `TICKER` je řetězec neobsahující bílé znaky a znak středníku
- `TYP` `TRANSAKCE` nabývá hodnoty `buy` nebo `sell` značící nákup resp. prodej
- `JEDNOTKOVÁ CENA` je cena za jednu akcii, jednotku komodity, atp. s přesností na maximálně dvě desetinná místa; jako oddělovač jednotek a desetín slouží znak tečky `.`; Např. `1234567.89`
- `MENA` je třípísmenný kód měny, např. `USD`, `EUR`, `CZK`, `SEK`, `GBP` atd.
- `OBJEM` značí množství jednotek (akcií, jednotek komodity atp.) v transakci
- `ID` je identifikátor transakce (řetězec neobsahující bílé znaky a znak středníku)

*Hodnota transakce* je `JEDNOTKOVÁ CENA * OBJEM`. Příklad záznamů:

```
2021-07-29 23:43:13;TSM;buy;667.98;USD;386;65fb53f6-7943-11eb-88cb-8c85986a186d
2021-07-29 23:43:15;BTC;sell;50100;USD;5;65467d26-7943-11eb-88cb-8c85986a186d
```

- První záznam značí nákup 306 akcii firmy TSMC (ticker `TSM`) za cenu 667.90 USD / akcie. Hodnota transakce je tedy 204377.40 USD.
  - Druhý záznam značí prodej 5 bitcoinů (ticker `BTC`) za cenu 50 100 USD / bitcoin. Hodnota transakce je tedy 250500.00 USD.
- Předpokládejte, že měna je u všech záznamů stejná (není potřeba ověřovat).
  - Skript žádný soubor nemodifikuje. Skript nepoužívá dočasné soubory.
  - Můžete předpokládat, že záznamy jsou ve vstupních souborech uvedeny chronologicky a je-li na vstupu více souborů, jejich pořadí je také chronologické.
  - Celkový zisk z uzavřených pozic* (příkaz `profit`) se spočítá jako *suma hodnot sell transakcí* - *suma hodnot buy transakcí*.
  - Hodnota aktuálně držených pozic* (příkazy `pos` a `graph-pos`) se pro každý ticker spočítá jako *počet držených jednotek \* jednotková cena z poslední transakce*, kde *počet držených jednotek* je dán jako *suma objemů buy transakcí* - *suma objemů sell transakcí*.
  - Pokud není při použití příkazu `hist-ord` uvedena šířka `WIDTH`, pak každá pozice v histogramu odpovídá jedné transakci.
  - Pokud není při použití příkazu `graph-pos` uvedena šířka `WIDTH`, pak každá pozice v histogramu odpovídá hodnotě 1000 (zaokrouhleno na tisíce směrem k nule, tj. hodnota 2000 bude reprezentována jako `##` zatímco hodnota 1999.99 jako `#` a hodnota -1999.99 jako `!`.
  - U příkazů `hist-ord` a `graph-pos` s uvedenou šířkou `WIDTH` při dělení zaokrouhluje směrem k nule (tedy např. při `graph-pos -w 6` a nejdelším řádku s hodnotou 1234 bude řádek s hodnotou 1234 vypadat takto `#####`, řádek s hodnotou 1233.99 takto `#####` a řádek s hodnotou -1233.99 takto `!!!!!!`).
  - Pořadí argumentů stačí uvažovat takové, že nejlví budou všechny přepínače, pak (volitelně) příkaz a nakonec seznam vstupních souborů (že tedy použít `getopts`). Podpora argumentů v libovolném pořadí je nepovinné rozšíření, jehož implementace může kompenzovat případnou ztrátu bodů v jiné části projektu.
  - Předpokládejte, že vstupní soubory nemůžou mít jména odpovídající některému příkazu nebo přepínači.
  - V případě uvedení přepínače `-h` nebo `--help` se vždy pouze vypíše nápověda a skript skončí (tedy, pokud by za přepínačem následoval nějaký příkaz nebo soubor, neprovede se).
  - Při výpisu pomocí příkazů `pos`, `last-price`, `hist-ord` a `graph-pos` musí být tickery zarovnaný doleva a dvojtečka na 11. pozici na řádku (výplň proveďte pomocí mezer). U příkazů `hist-ord` a `graph-pos` je za dvojtečkou na všech řádcích právě jedna mezeira (případně žádná, pokud v pravém sloupci daného řádku nic není), u příkazů `pos` a `last-price` jsou hodnoty v pravé části výpisu formátovány tak, aby (v případě neprázdného výpisu) byla na řádku s nejdelší řetězcovou reprezentací hodnoty (tj. včetně znaménka) mezi dvojtečkou a hodnotou právě jedna mezeira a ostatní řádky byly zarovnaný doprava vzhledem k délce tohoto řádku (vizte příklady výpisů níže).

### Návratová hodnota

- Skript vrací úspěch v případě úspěšné operace. Interní chyba skriptu nebo chybné argumenty budou doprovázeny chybovým hlášením a neúspěšným návratovým kódem.

### Implementační detaily

- Skript by měl mít v celém běhu nastaveno `POSIXLY_CORRECT=yes`.
- Skript by měl běžet na všech běžných shellech (`dash`, `ksh`, `bash`). Pokud použijete vlastnost specifikou pro nějaký shell, uveďte to pomocí direktivy interpretu na prvním řádku souboru, např. `#!/bin/bash` nebo `#!/usr/bin/env bash` pro `bash`. Můžete použít GNU rozšíření pro `sed` či `awk`. Jazyky Perl, Python, Ruby, atd. povoleny nejsou.
- UPOZORNĚNÍ:** některé servery, např. `merlin.fit.vutbr.cz`, mají symlink `/bin/sh -> bash`. Ověřte si proto, že skript skutečně testujete daným shellem. Doporučuji ověřit správnou funkčnost pomocí virtuálního stroje níže.
- Skript musí běžet na běžně dostupných OS GNU/Linux, BSD a MacOS. Studentům je k dispozici virtuální stroj s obrazem ke stažení zde: <http://www.fit.vutbr.cz/~lengal/public/trusty.ova> (pro VirtualBox, login: `trusty` / heslo: `trusty`), na kterém lze ověřit správnou funkčnost projektu.
- Skript nesmí používat dočasné soubory. Povoleny jsou však dočasné soubory nepřímo tvořené jinými příkazy (např. příkazem `sed -i`).
- Čísla vypisuje v desítkovém zápisu s přesností na dvě desetinná místa. Pozor, některé nástroje (např. `awk`) mohou větší čísla vypisovat implicitně pomocí vědeckého zápisu.

### Odevzdání projektu

Odevzdávejte pouze skript `tradelog` (nebalte ho do žádného archivu). Odevzdejte do IS, termín Projekt 1.

### Rady

- Dobrá dekompozice problému na podproblémy Vám může značně ulehčit práci a předejít chybám.
- Naučte se *dobře* používat funkce v shellu

### Příklady použití

- Úkazy záznamů nástroje pro obchodování na burze jsou dostupné zde: <https://pajda.fit.vutbr.cz/ios/ios-21-1-logs>

```
Příklady:

$ cat stock-2.log | head -n 5 | ./tradelog
2021-07-29 15:30:42;MSFT;sell;240.07;USD;327;65fa0854-7943-11eb-9294-8c85986a186d
2021-07-29 15:31:12;MA;sell;314.91;USD;712;65fae24a-7943-11eb-9171-8c85986a186d
2021-07-29 15:31:32;BAC;buy;34.16;USD;635;65fae466-7943-11eb-8f48-8c85986a186d
2021-07-29 15:37:09;BAC;sell;36.67;USD;897;65fae614-7943-11eb-9c0b-8c85986a186d
2021-07-29 15:43:02;JPM;sell;146.77;USD;190;65fae79a-7943-11eb-8977-8c85986a186d

$ ./tradelog -t TSLA -t V stock-2.log
2021-07-29 17:06:57;TSLA;buy;757.57;USD;812;65farb04-7943-11eb-8d41-8c85986a186d
2021-07-29 17:58:16;V;sell;215.31;USD;406;65fb0862-7943-11eb-87fe-8c85986a186d
2021-07-29 18:12:27;TSLA;sell;729.75;USD;482;65fb0892-7943-11eb-867f-8c85986a186d
2021-07-29 18:55:19;V;sell;217.92;USD;218;65fb1238-7943-11eb-86e2-8c85986a186d
2021-07-29 19:19:26;TSLA;sell;700.75;USD;457;65fb1792-7943-11eb-8abf-8c85986a186d
2021-07-29 19:27:39;TSLA;buy;710.79;USD;633;65fb1908-7943-11eb-a5d9-8c85986a186d
2021-07-29 20:06:53;V;sell;210.72;USD;272;65fb237c-7943-11eb-83a9-8c85986a186d
2021-07-29 20:59:16;V;sell;196.54;USD;92;65fb2c32-7943-11eb-9dd3-8c85986a186d
2021-07-29 21:03:15;V;buy;188.60;USD;605;65fb2d4a-7943-11eb-8804-8c85986a186d
2021-07-29 21:17:37;V;sell;222.52;USD;447;65fb2f7a-7943-11eb-8f28-8c85986a186d
2021-07-29 21:18:18;TSLA;buy;733.96;USD;720;65fb3092-7943-11eb-992a-8c85986a186d
2021-07-29 21:50:25;V;sell;212.58;USD;203;65fb3a2e-7943-11eb-8e0b-8c85986a186d
2021-07-29 22:10:55;TSLA;sell;718.31;USD;3794;65fb3f88-7943-11eb-a371-8c85986a186d
2021-07-29 22:21:31;TSLA;sell;681.74;USD;7122;65fb41a4-7943-11eb-a09f-8c85986a186d
2021-07-29 23:01:47;TSLA;sell;707.03;USD;1578;65fb4a50-7943-11eb-9f6e-8c85986a186d
2021-07-29 23:21:11;TSLA;buy;679.27;USD;9655;65fb4fb4-7943-11eb-8199-8c85986a186d
2021-07-29 23:43:13;TSLA;buy;607.90;USD;306;65fb53f6-7943-11eb-88cb-8c85986a186d
2021-07-29 23:48:29;V;buy;195.52;USD;2003;65fb5824-7943-11eb-9b59-8c85986a186d

$ ./tradelog -t CVX stock-4.log.gx | head -n 3
2021-09-27 05:12:30;CVX;sell;1008.17;USD;98;8f229a62-7945-11eb-a6fb-8c85986a186d
2021-09-27 13:57:48;CVX;sell;94.81;USD;5374;8f22ec38-7945-11eb-8c68-8c85986a186d
2021-09-27 14:52:50;CVX;sell;89.22;USD;7759;8f22f46c-7945-11eb-9bb2-8c85986a186d

$ ./tradelog list-tick stock-2.log
AAPL
ANZN
BABA
BAC
DIS
FB
GOOG
GOOGL
JNJ
JPM
MA
MSFT
NVDA
PG
PYPL
TSLA
TSM
UNH
V
WMT

$ ./tradelog profit stock-2.log
-58863165.03

$ ./tradelog -t TSM -t PYPL profit stock-2.log
-577302.62

$ ./tradelog pos stock-2.log
ANZN      : 64645275.64
GOOGL     : 7914309.08
NVDA      : 2540507.69
DIS       : 1925621.88
TSM       : 1266217.38
JPM       : 937220.31
BABA      : 444692.64
BAC       : 323899.29
JNJ       : 81769.32
FB        : 42673.05
WMT       : 2423.34
MSFT      : -321051.04
V         : -322999.04
PYPL      : -502892.46
MA        : -569746.42
TSLA      : -872945.30
PG        : -1138085.10
AAPL      : -1190996.48
UNH       : -181240.88
GOOG      : -9846258.51

$ ./tradelog -t TSM -t PYPL -t AAPL pos stock-2.log
TSM       : 1266217.38
PYPL      : -502892.46
AAPL      : -1190996.48

$ ./tradelog last-price stock-2.log
AAPL      : 133.88
ANZN      : 3496.04
BABA      : 245.28
BAC       : 38.61
DIS       : 207.48
FB        : 275.31
GOOG      : 1975.97
GOOGL     : 1990.04
JNJ       : 155.16
JPM       : 135.77
MA        : 333.38
MSFT      : 237.64
NVDA      : 629.93
PG        : 124.70
PYPL      : 279.54
TSLA      : 667.90
TSM       : 140.41
UNH       : 321.06
V         : 195.52
WMT       : 134.63

$ ./tradelog hist-ord stock-2.log
AAPL      : ##
ANZN      : #####
BABA      : #####
BAC       : #####
DIS       : #####
FB        : #####
GOOG      : #####
GOOGL     : #####
JNJ       : ##
JPM       : #####
MA        : #####
MSFT      : #####
NVDA      : #####
PG        : #####
PYPL      : #####
TSLA      : #####
TSM       : ##
UNH       : ##
V         : #####
WMT       : #####

$ ./tradelog -w 100 graph-pos stock-6.log
AAPL      : |
ANZN      : |
BABA      : |
BAC       : |
DIS       : |
FB        : |
GOOG      : |
GOOGL     : |
JNJ       : |
JPM       : |
MA        : |
MSFT      : |
NVDA      : |
PG        : |
PYPL      : |
TSLA      : |
TSM       : |
UNH       : |
V         : |
WMT       : |

$ ./tradelog -w 10 -t FB -t JNJ -t WMT graph-pos stock-6.log
FB        : #####
JNJ       : #####
WMT       : |

$ cat /dev/null | ./tradelog profit
0.00
```