

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ

Отчёт по утилите

по курсу «Системное программирование»

по теме «Исследование утилиты df»

Выполнил студент гр. 13541/2:
Волкова М.Д.

Проверил преподаватель:
Душутина Е. В.

Санкт-Петербург
2019 г.

1 Цель работы

Цель работы выполнить техническое задание.

2 Техническое задание

Необходимо познакомиться с утилитой `df`, а также исследовать её и выполнить заданную модификацию. Исследование утилиты `df` заключается в выполнении всех пунктов, представленных ниже:

- понять для чего необходима данная утилита
- рассмотреть все особенности утилиты
- запустить утилиту с всевозможными ключами
- выполнить анализ исходного кода
- проанализировать источник данных утилиты
- определить за что отвечает каждый участок кода, выполнить декомпозицию

3 Характеристики системы

```
1 lorismelik@lorismelik-Aspire-Z5700:~/$ cat /proc/version
2 Linux version 4.15.0-29-generic (buildd@lgw01-amd64-057) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3))
   #31-Ubuntu SMP Tue Jul 17 15:39:52 UTC 2018
3
4 lorismelik@lorismelik-Aspire-Z5700:~/$ gcc --version
5 gcc (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0
6 Copyright (C) 2017 Free Software Foundation, Inc.
7 This is free software; see the source for copying conditions. There is NO
8 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
9
10 lorismelik@lorismelik-Aspire-Z5700:~/$ strace -V
11 strace -- version UNKNOWN
12 Copyright (c) 1991-2018 The strace developers <https://strace.io>.
13 This is free software; see the source for copying conditions. There is NO
14 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
15
16 Optional features enabled: stack-unwind stack-demangle m32-mpers mx32-mpers
17
18 lorismelik@lorismelik-Aspire-Z5700:~/$ ltrace -V
19 ltrace version 0.7.3.
20 Copyright (C) 1997-2009 Juan Cespedes <cespedes@debian.org>.
21 This is free software; see the GNU General Public Licence
22 version 2 or later for copying conditions. There is NO warranty.
```

4 Что такое df?

Для того, чтобы посмотреть, сколько места занимает файловая система, диск, файл или каталог, в Linux существует утилита: `df`. `Df` (disk free) выводит список всех файловых систем по именам устройств с указанием размера, показывает точки монтирования и количество свободного/занятого пространства.

Утилита `df` поставляется по умолчанию во всех дистрибутивах Linux и имеет очень простой синтаксис.

```
1 $ df option device
```

Существует 2 версии: POSIX и GNU

- POSIX По умолчанию все размеры выдаются в блоках по 512 байт. Если файл является не обычным файлом, а каталогом или FIFO, результат не определён.
- GNU По умолчанию все размеры выдаются в блоках по 1024-байт. Если аргумент файл является дисковым файлом устройства, содержащим смонтированную файловую систему, то `df` показывает доступное пространство на этой файловой системе, а не на той, где содержится файл устройства. Намного большее количество ключей по сравнению с первой версией.

5 Поддерживаемые опции

- `-h`, `--human-readable` Отобразит размер в человеко-читаемом формате, добавив названия единиц (Килобайт, Мегабайт, Гигабайт, Терабайт). На рисунке далее показан вывод утилиты. В первом столбце указана файловая система. Во втором общий размер памяти в ней. В третьем и четвертом указано кол-во использованной и оставшейся памяти. В четвертом столбце указано кол-во использованной памяти в процентах. В последнем столбце указана точка монтирования системы (входная точка к файлам этой системы).

```
lorismelik@lorismelik-Aspire-Z5700:/home$ df -h
Файл.система  Размер  Использовано  Дост  Использовано%  Смонтировано в
udev          3,9G    2,2M    3,9G    1% /dev
tmpfs         794M    1,9M    792M    1% /run
/dev/sda5     92G     8,6G    78G     10% /
tmpfs         3,9G    40M     3,9G    1% /dev/shm
tmpfs         5,0M    4,0K    5,0M    1% /run/lock
tmpfs         3,9G    0       3,9G    0% /sys/fs/cgroup
/dev/loop0    141M    141M    0       100% /snap/gnome-3-26-1604/70
/dev/loop2    87M     87M    0       100% /snap/core/4917
/dev/loop3    35M     35M    0       100% /snap/gtk-common-themes/818
/dev/loop5    141M    141M    0       100% /snap/gnome-3-26-1604/74
/dev/loop6    3,8M    3,8M    0       100% /snap/gnome-system-monitor/51
/dev/loop8    2,4M    2,4M    0       100% /snap/gnome-calculator/180
/dev/loop1    13M     13M    0       100% /snap/gnome-characters/103
/dev/loop4    15M     15M    0       100% /snap/gnome-logs/37
/dev/loop9    99M     99M    0       100% /snap/telegram-desktop/291
/dev/loop10   13M     13M    0       100% /snap/gnome-characters/139
/dev/loop7    35M     35M    0       100% /snap/gtk-common-themes/319
/dev/loop11   3,8M    3,8M    0       100% /snap/gnome-system-monitor/57
/dev/loop12   15M     15M    0       100% /snap/gnome-logs/45
/dev/loop13   89M     89M    0       100% /snap/core/5897
tmpfs         794M    56K     794M    1% /run/user/1000
```

Рис. 1: `-h`

- -a, -all Включает в список файловых систем те, которые имеют размер в 0 блоков, и которые по умолчанию опускаются.

```

lorismelik@lorismelik-Aspire-Z5700:/home$ df -a
Файл.система  1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
sysfs          0           0           0           - /sys
proc           0           0           0           - /proc
udev          4032888     1992        4030896     1% /dev
devpts         0           0           0           - /dev/pts
tmpfs          812632     1892        810740     1% /run
/dev/sda5      95599012   10821576    79878160    12% /
securityfs     0           0           0           - /sys/kernel/security
tmpfs          4063152     96688       3966464     3% /dev/shm
tmpfs          5120        4           5116        1% /run/lock
tmpfs          4063152     0           4063152     0% /sys/fs/cgroup
cgroup         0           0           0           - /sys/fs/cgroup/unified
cgroup         0           0           0           - /sys/fs/cgroup/systemd
pstore         0           0           0           - /sys/fs/pstore
cgroup         0           0           0           - /sys/fs/cgroup/cpu,cpuacct
cgroup         0           0           0           - /sys/fs/cgroup/cpuset
cgroup         0           0           0           - /sys/fs/cgroup/net_cls,net_prio
cgroup         0           0           0           - /sys/fs/cgroup/freezer
cgroup         0           0           0           - /sys/fs/cgroup/blkio
cgroup         0           0           0           - /sys/fs/cgroup/hugetlb
cgroup         0           0           0           - /sys/fs/cgroup/perf_event
cgroup         0           0           0           - /sys/fs/cgroup/memory
cgroup         0           0           0           - /sys/fs/cgroup/rdma
cgroup         0           0           0           - /sys/fs/cgroup/devices
cgroup         0           0           0           - /sys/fs/cgroup/pids
systemd-1      0           0           0           - /proc/sys/fs/binfmt_misc
mqueue         0           0           0           - /dev/mqueue
debugfs        0           0           0           - /sys/kernel/debug
hugetlbfs      0           0           0           - /dev/hugepages
tracefs        -           -           -           - /sys/kernel/debug/tracing

```

Рис. 2: -a

- -k Используется для отображения размеров блоками по 1 килобайту, вместо установленных по умолчанию блоков в 512 байт.
- -i, -inodes Вместо информации о блоках выдаётся информация об использовании inode'ов в файловой системе. Inode содержит информацию о файле: владелец, права доступа, временные штампы и местоположение на диске.

```
lorismelik@lorismelik-Aspire-Z5700:/home$ df -i
Файл.система      Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
udev              1008222      602    1007620          1% /dev
tmpfs             1015788     1041    1014747          1% /run
/dev/sda5         6111232    209885    5901347          4% /
tmpfs             1015788      102    1015686          1% /dev/shm
tmpfs             1015788        5    1015783          1% /run/lock
tmpfs             1015788       18    1015770          1% /sys/fs/cgroup
/dev/loop1        12808      12808         0        100% /snap/core/5897
/dev/loop2        12810      12810         0        100% /snap/core/6130
/dev/loop0         1720      1720         0        100% /snap/gnome-logs/37
/dev/loop5        26698      26698         0        100% /snap/gtk-common-themes/319
/dev/loop3         1597      1597         0        100% /snap/gnome-characters/103
/dev/loop4        27631      27631         0        100% /snap/gnome-3-26-1604/74
/dev/loop7        27638      27638         0        100% /snap/gnome-3-26-1604/70
/dev/loop8         1269      1269         0        100% /snap/gnome-calculator/260
/dev/loop10        724       724         0        100% /snap/gnome-system-monitor/51
/dev/loop11        1272      1272         0        100% /snap/gnome-calculator/180
/dev/loop9        27345      27345         0        100% /snap/gtk-common-themes/818
/dev/loop12        9884      9884         0        100% /snap/telegram-desktop/490
/dev/loop14        1598      1598         0        100% /snap/gnome-characters/139
/dev/loop13       23655      23655         0        100% /snap/telegram-desktop/291
/dev/loop15        1720      1720         0        100% /snap/gnome-logs/45
/dev/loop16        747       747         0        100% /snap/gnome-system-monitor/57
tmpfs             1015788       55    1015733          1% /run/user/1000
/dev/sda1        316524972  1254910  315270062          1% /media/lorismelik/8042717E42717A2C
/dev/loop17        12814      12814         0        100% /snap/core/6259
/dev/sdb1         95491528   786592   94704936          1% /media/lorismelik/E48A74438A741470
/dev/sda2        190502892   637925  189864967          1% /media/lorismelik/9A0664000663DBAF
```

Рис. 3: -и

- -t тип-файловой-системы, -type=тип-файловой-системы Показывать только файловые системы с указанным типом-файловой-системы.

```
lorismelik@lorismelik-Aspire-Z5700:/home$ df -t tmpfs
```

Файл.система	1K-блоков	Использовано	Доступно	Использовано%	Смонтировано в
tmpfs	812632	1892	810740	1%	/run
tmpfs	4063152	98520	3964632	3%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	4063152	0	4063152	0%	/sys/fs/cgroup
tmpfs	812628	100	812528	1%	/run/user/1000

Рис. 4: -t

- -T, --print-type Выдавать тип для каждой файловой системы. Тип берётся от операционной системы (и определяется системно-зависимым способом, например посредством чтения файла /etc/mtab).

```
lorismelik@lorismelik-Aspire-Z5700:/home$ df -T
```

Файл.система	Тип	1К-блоков	Использовано	Доступно	Использовано%	Смонтировано в
/dev	devtmpfs	4032888	1992	4030896	1%	/dev
tmpfs	tmpfs	812632	1892	810740	1%	/run
/dev/sda5	ext4	95599012	10822276	79877460	12%	/
tmpfs	tmpfs	4063152	97312	3965840	3%	/dev/shm
tmpfs	tmpfs	5120	4	5116	1%	/run/lock
tmpfs	tmpfs	4063152	0	4063152	0%	/sys/fs/cgroup
/dev/loop1	squashfs	90368	90368	0	100%	/snap/core/5897
/dev/loop2	squashfs	91648	91648	0	100%	/snap/core/6130
/dev/loop0	squashfs	14848	14848	0	100%	/snap/gnome-logs/37
/dev/loop5	squashfs	35584	35584	0	100%	/snap/gtk-common-themes/319
/dev/loop3	squashfs	13312	13312	0	100%	/snap/gnome-characters/103
/dev/loop4	squashfs	144128	144128	0	100%	/snap/gnome-3-26-1604/74
/dev/loop7	squashfs	144384	144384	0	100%	/snap/gnome-3-26-1604/70
/dev/loop8	squashfs	2304	2304	0	100%	/snap/gnome-calculator/260
/dev/loop10	squashfs	3840	3840	0	100%	/snap/gnome-system-monitor/51
/dev/loop11	squashfs	2432	2432	0	100%	/snap/gnome-calculator/180
/dev/loop9	squashfs	35456	35456	0	100%	/snap/gtk-common-themes/818
/dev/loop12	squashfs	96000	96000	0	100%	/snap/telegram-desktop/490
/dev/loop14	squashfs	13312	13312	0	100%	/snap/gnome-characters/139
/dev/loop13	squashfs	101376	101376	0	100%	/snap/telegram-desktop/291
/dev/loop15	squashfs	14976	14976	0	100%	/snap/gnome-logs/45
/dev/loop16	squashfs	3840	3840	0	100%	/snap/gnome-system-monitor/57
tmpfs	tmpfs	812628	108	812520	1%	/run/user/1000
/dev/sda1	fuseblk	488379388	173165136	315214252	36%	/media/lorismelik/8042717E427
/dev/loop17	squashfs	93312	93312	0	100%	/snap/core/6259
/dev/sdb1	fuseblk	234428412	139788852	94639560	60%	/media/lorismelik/E48A74438A7
/dev/sda2	fuseblk	390722656	200907892	189814764	52%	/media/lorismelik/9A066400066

Рис. 5: -T

- -x тип-файловой-системы, --exclude-type=тип-файловой-системы Не показывать файловые системы с заданным типом-файловой-системы.

Стандартные опции GNU:

- --help Выдать подсказку на стандартный вывод и успешно завершиться.
- --version Выдать информацию о версии на стандартный вывод и успешно завершиться
- -- Служит для обозначения конца списка опций.

6 Структура исходного кода

Весь код утилиты находится в одном файле df.c и имеет порядка 1800 строк. Но сама утилита находится в пакете coreutils от GNU и задействует некоторые другие файлы из этого пакета в своей работе, поэтому для сборки потребуется весь пакет, который собирается с помощью makefile.

7 Принцип работы утилиты

Большую часть main функции составляет интерфейсная часть, в ней обрабатываются опции и флаги поданные на вход. Остальная же функциональность разбита на функции. Можно выделить вспомогательные функции, функций для вывода информации пользователю и функции для получения системной информации (основные).

Получения системной информации:

- get_dev - Возвращает информацию о устройстве.
- get_point - Если директория или файл устройства смонтирован, то показывает его использование диска. Для получения информации используется вызов stat

- `get_all_entries`-показывает все смонтированные файловые системы, кроме пустых.

В функции `get_dev` используется функция `get_fs_usage` для получения всей информации. Она также находится в пакете `coreutils`. Эта функция в свою очередь использует системную функцию `statvfs`, которая возвращает данные о смонтированной файловой системе.

Структура `statvfs` содержит в себе всю необходимую информацию: кол-во свободных и использованных блоков и индексных дескрипторов (`inode`).

```
struct statvfs {
    unsigned long  f_bsize;      /* file system block size */
    unsigned long  f_frsize;     /* fragment size */
    fsblkcnt_t     f_blocks;     /* size of fs in f_frsize units */
    fsblkcnt_t     f_bfree;      /* # free blocks */
    fsblkcnt_t     f_bavail;     /* # free blocks for unprivileged users */
    fsfilcnt_t     f_files;      /* # inodes */
    fsfilcnt_t     f_ffree;      /* # free inodes */
    fsfilcnt_t     f_favail;     /* # free inodes for unprivileged users */
    unsigned long  f_fsid;       /* file system ID */
    unsigned long  f_flag;       /* mount flags */
    unsigned long  f_namemax;    /* maximum filename length */
};
```

Рис. 6: `statvfs`

Для получения информации о файле используется системный вызов stat. Эти функции возвращают информацию об указанном файле. Для этого не требуется иметь права доступа к файлу, хотя потребуются права поиска во всех каталогах, указанных в полном имени файла. На рисунке показана структура, которую использует программа как ответ на вызов stat. В ней указана вся необходимая информация для работы утилиты.

```
struct stat {  
    dev_t      st_dev;      /* устройство */  
    ino_t      st_ino;      /* inode */  
    mode_t     st_mode;     /* режим доступа */  
    nlink_t    st_nlink;    /* количество жестких ссылок */  
    uid_t      st_uid;      /* идентификатор пользователя-владельца */  
    gid_t      st_gid;      /* идентификатор группы-владельца */  
    dev_t      st_rdev;     /* тип устройства */  
    /* (если это устройство) */  
    off_t      st_size;     /* общий размер в байтах */  
    blksize_t  st_blksize;  /* размер блока ввода-вывода */  
    /* в файловой системе */  
    blkcnt_t   st_blocks;   /* количество выделенных блоков */  
    time_t     st_atime;    /* время последнего доступа */  
    time_t     st_mtime;    /* время последней модификации */  
    time_t     st_ctime;    /* время последнего изменения */  
};
```

Рис. 7: stat

Далее предоставлен код функции `get_point`, которая определяет `*point` это устройство или директория и вычисляет свободное место.

```

1 static void
2 get_point (const char *point, const struct stat *statp)
3 {
4     struct stat disk_stats;
5     struct mount_entry *me;
6     struct mount_entry const *best_match = NULL;
7
8     /* Calculate the real absolute file name for POINT, and use that to find
9      the mount point. This avoids statting unavailable mount points,
10     which can hang df. */
11     char *resolved = canonicalize_file_name (point);
12     if (resolved && resolved[0] == '/')
13     {
14         size_t resolved_len = strlen (resolved);
15         size_t best_match_len = 0;
16
17         for (me = mount_list; me; me = me->me_next)
18         {
19             if (!STREQ (me->me_type, "lofs")
20                 && (!best_match || best_match->me_dummy || !me->me_dummy))
21             {
22                 size_t len = strlen (me->me_mountdir);
23                 if (best_match_len <= len && len <= resolved_len
24                     && (len == 1 /* root file system */
25                         || ((len == resolved_len || resolved[len] == '/')
26                             && STREQ_LEN (me->me_mountdir, resolved, len))))
27                 {
28                     best_match = me;
29                     best_match_len = len;
30                 }
31             }
32         }
33     }
34     free (resolved);
35     if (best_match
36         && (stat (best_match->me_mountdir, &disk_stats) != 0
37             || disk_stats.st_dev != statp->st_dev))
38         best_match = NULL;
39
40     if (! best_match)
41         for (me = mount_list; me; me = me->me_next)
42         {
43             if (me->me_dev == (dev_t) -1)
44             {
45                 if (stat (me->me_mountdir, &disk_stats) == 0)
46                     me->me_dev = disk_stats.st_dev;
47                 else
48                 {
49                     /* Report only I/O errors. Other errors might be
50                      caused by shadowed mount points, which means POINT
51                      can't possibly be on this file system. */
52                     if (errno == EIO)
53                     {
54                         error (0, errno, "%s", quotef (me->me_mountdir));
55                         exit_status = EXIT_FAILURE;
56                     }
57
58                     /* So we won't try and fail repeatedly. */
59                     me->me_dev = (dev_t) -2;
60                 }
61             }
62
63             if (statp->st_dev == me->me_dev
64                 && !STREQ (me->me_type, "lofs")
65                 && (!best_match || best_match->me_dummy || !me->me_dummy))
66             {
67                 /* Skip bogus mtab entries. */
68                 if (stat (me->me_mountdir, &disk_stats) != 0
69                     || disk_stats.st_dev != me->me_dev)
70                     me->me_dev = (dev_t) -2;
71                 else
72                     best_match = me;
73             }
74         }
75
76     if (best_match)
77         get_dev (best_match->me_devname, best_match->me_mountdir, point, point,
78                 best_match->me_type, best_match->me_dummy, best_match->me_remote,
79                 NULL, false);
80     else
81     {
82         /* We couldn't find the mount entry corresponding to POINT. Go ahead and
83          print as much info as we can; methods that require the device to be
84          present will fail at a later point. */
85
86         /* Find the actual mount point. */
87         char *mp = find_mount_point (point, statp);

```

```

88     if (mp)
89     {
90         get_dev (NULL, mp, point, NULL, NULL, false, false, NULL, false);
91         free (mp);
92     }
93 }
94 }

```

Здесь показана структура, которая определяет как будет выглядеть выходная таблица:

```

1 static struct field_data_t field_data[] = {
2     [SOURCE_FIELD] = { SOURCE_FIELD,
3         "source", OTHER_FLD, N_("Filesystem"), 14, MBS_ALIGN_LEFT, false },
4
5     [FSTYPE_FIELD] = { FSTYPE_FIELD,
6         "fstype", OTHER_FLD, N_("Type"), 4, MBS_ALIGN_LEFT, false },
7
8     [SIZE_FIELD] = { SIZE_FIELD,
9         "size", BLOCK_FLD, N_("blocks"), 5, MBS_ALIGN_RIGHT, false },
10
11     [USED_FIELD] = { USED_FIELD,
12         "used", BLOCK_FLD, N_("Used"), 5, MBS_ALIGN_RIGHT, false },
13
14     [AVAIL_FIELD] = { AVAIL_FIELD,
15         "avail", BLOCK_FLD, N_("Available"), 5, MBS_ALIGN_RIGHT, false },
16
17     [PCENT_FIELD] = { PCENT_FIELD,
18         "pcent", BLOCK_FLD, N_("Use%"), 4, MBS_ALIGN_RIGHT, false },
19
20     [ITOTAL_FIELD] = { ITOTAL_FIELD,
21         "itotal", INODE_FLD, N_("Inodes"), 5, MBS_ALIGN_RIGHT, false },
22
23     [IUSED_FIELD] = { IUSED_FIELD,
24         "iused", INODE_FLD, N_("IUsed"), 5, MBS_ALIGN_RIGHT, false },
25
26     [IAVAIL_FIELD] = { IAVAIL_FIELD,
27         "iavail", INODE_FLD, N_("IFree"), 5, MBS_ALIGN_RIGHT, false },
28
29     [IPCENT_FIELD] = { IPCENT_FIELD,
30         "ipcent", INODE_FLD, N_("IUse%"), 4, MBS_ALIGN_RIGHT, false },
31
32     [TARGET_FIELD] = { TARGET_FIELD,
33         "target", OTHER_FLD, N_("Mounted on"), 0, MBS_ALIGN_LEFT, false },
34
35     [FILE_FIELD] = { FILE_FIELD,
36         "file", OTHER_FLD, N_("File"), 0, MBS_ALIGN_LEFT, false }
37 };

```

8 Вывод

В ходе выполнения данной работы познакомился с утилитой `df`. Также была проверена работа всевозможных ключей и флагов утилиты. Кроме того был исследован исходный код утилиты. Прежде всего были внесены некоторые комментарии в исходный код для более понятного чтения. В итоге хотелось бы отметить то, что `df` – имеет вполне читаемый код, но данный момент существуют более удобные утилиты, например, `discus`.