

## Monitorowanie obciążenia systemu

### 1. Weryfikacja aktualnego obciążenia systemu przez poszczególne procesy działający w konsoli systemu Unix - aplikacja htop

Instalacja pakietu:

```
apt-get install htop
```

Zastosowanie polecenia htop: Wyświetlane dane przedstawiają zużycie mocy obliczeniowej procesora, ilości zajmowanej pamięci RAM, czasu działania, numeru procesu itd. htop możemy użyć w zastępstwie za systemowego top'a, tylko w przypadku htop wyniki prezentowane są w sposób przejrzystszy dla użytkownika.

```

 1  [|||||] 12.9% Tasks: 132, 10 thr; 1 running
 2  [ 0.0% Load average: 0.67 0.79 0.89
 3  [|| 1.4% Uptime: 38 days, 10:21:43
 4  [||||| 5.7%
 5  [|| 0.5%
 6  [|| 0.5%
 7  [ 0.0%
 8  [|| 0.5%
Mem[|||||] [2164/16085MB]
Swp[|||||] 889/3811MB]

```

PID	USER	PRI	NI	VRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
8009	www-data	20	0	314M	38936	11992	S	18.1	0.2	0:02.18	/usr/sbin/apache2 -k start
1591	postgres	20	0	3905M	194M	171M	S	6.7	1.2	0:21.41	postgres: 9.6/main: httpd
8230	www-data	20	0	303M	35328	17104	S	4.3	0.2	0:00.41	/usr/sbin/apache2 -k start
8259	www-data	20	0	303M	27012	11084	S	4.3	0.2	0:00.38	/usr/sbin/apache2 -k start
8101	www-data	20	0	305M	31920	16208	S	2.4	0.2	0:00.45	/usr/sbin/apache2 -k start
7009	www-data	20	0	303M	28188	12080	S	1.9	0.2	0:02.76	/usr/sbin/apache2 -k start
8005	www-data	20	0	314M	37936	11544	S	1.4	0.2	0:00.50	/usr/sbin/apache2 -k start
8234	www-data	20	0	302M	24996	10584	S	1.4	0.2	0:00.23	/usr/sbin/apache2 -k start
8232	www-data	20	0	303M	27024	12012	S	1.0	0.2	0:00.78	/usr/sbin/apache2 -k start
8308	www-data	20	0	303M	25500	10392	S	1.0	0.2	0:00.13	/usr/sbin/apache2 -k start
8100	www-data	20	0	303M	27176	11632	S	0.5	0.2	0:00.46	/usr/sbin/apache2 -k start
877	memcache	20	0	1187M	148M	2088	S	0.5	0.9	26:15.18	/usr/bin/memcached -m 4096 -I 20m -p 11211 -u memcache -l 127.0.0.1
8591	root	20	0	25812	4924	2888	R	0.5	0.0	0:00.04	htop
934	memcache	20	0	1187M	148M	2088	S	0.5	0.9	5:39.72	/usr/bin/memcached -m 4096 -I 20m -p 11211 -u memcache -l 127.0.0.1
8231	www-data	20	0	303M	26508	11316	S	0.5	0.2	0:00.27	/usr/sbin/apache2 -k start
8233	www-data	20	0	310M	34216	11056	S	0.0	0.2	0:00.37	/usr/sbin/apache2 -k start
7444	www-data	20	0	303M	26732	11572	S	0.0	0.2	0:01.73	/usr/sbin/apache2 -k start
7842	www-data	20	0	310M	35672	12912	S	0.0	0.2	0:02.06	/usr/sbin/apache2 -k start
7843	www-data	20	0	304M	27160	10712	S	0.0	0.2	0:00.43	/usr/sbin/apache2 -k start
8342	www-data	20	0	301M	24524	10648	S	0.0	0.1	0:00.17	/usr/sbin/apache2 -k start
932	memcache	20	0	1187M	148M	2088	S	0.0	0.9	5:39.28	/usr/bin/memcached -m 4096 -I 20m -p 11211 -u memcache -l 127.0.0.1
933	memcache	20	0	1187M	148M	2088	S	0.0	0.9	5:39.67	/usr/bin/memcached -m 4096 -I 20m -p 11211 -u memcache -l 127.0.0.1
935	memcache	20	0	1187M	148M	2088	S	0.0	0.9	5:39.72	/usr/bin/memcached -m 4096 -I 20m -p 11211 -u memcache -l 127.0.0.1

### 2. Monitorowanie użycia pamięci trwałej - aplikacja iotop

Instalacja pakietu:

```
apt-get install iotop
```

Zastosowanie polecenia iotop: Idealnym zastosowaniem dla iotop jest monitorowanie własnej aplikacji - pewnego rodzaju test, debugowanie.

Można także obserwować które aplikacje najbardziej potrzebują pamięci trwałej aby móc takim aplikacjom stworzyć specjalne miejsce w pamięci ram z szybkim dostępem.

TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO>	COMMAND
28216	be/4	postgres	0.00 B/s	7.84 K/s	0.00 %	0.86 %	postgres: 9.6/main: http [local] idle
9724	be/4	postgres	0.00 B/s	7.84 K/s	0.00 %	0.28 %	postgres: 9.6/main: http [local] idle
10330	be/4	www-data	0.00 B/s	0.00 B/s	0.00 %	0.00 %	php /home/public_html/apps/backproc/thread_ignitor.php --threadId=t1 --parentId=703
702	be/4	www-data	0.00 B/s	0.00 B/s	0.00 %	0.00 %	php /home/public_html/apps/backproc/engine.php
3276	be/4	postgres	0.00 B/s	78.44 K/s	0.00 %	0.00 %	postgres: 9.6/main: checkpointer process
2416	be/4	www-data	0.00 B/s	0.00 B/s	0.00 %	0.00 %	php /home/public_html/apps/backproc/engine.php
2403	be/4	www-data	0.00 B/s	0.00 B/s	0.00 %	0.00 %	php /home/public_html/apps/backproc/engine.php
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	init
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kthreadd]
3	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/0]
5	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/0:0H]
7	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_sched]
8	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_bh]
9	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/0]
10	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/0]
11	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/1]
12	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/1]
13	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/1]
15	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/1:0H]
16	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/2]
17	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/2]
18	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/2]
531	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[xfs_mru_cache]
20	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/2:0H]

### 3. Weryfikacja parametrów dysków twardech hdparm

Instalacja pakietu:

```
apt-get install hdparm
```

Polecenia wykonujemy z poziomu roota. Przykłady użycia:

Listuje informacje bezpośrednio z urządzenia (lub z '-i' z proc)

```
hdparm -I /dev/sda
```

Włącza tryb DMA dla pierwszego dysku twardego.

```
hdparm -d1 /dev/sda
```

Wyłącza tryb DMA dla pierwszego dysku twardego.

```
hdparm -d0 /dev/sda
```

Sprawdza wydajność czytania pamięć cache dla pierwszego dysku twardego.

```
hdparm -T /dev/sda
```

Sprawdza wydajność czytania pierwszego dysku twardego.

```
hdparm -t /dev/sda
```

Zastosowanie polecenia hdparm

```
root@eDokumenty:/home/edokumenty# hdparm -tT /dev/sda1
/dev/sda1:
Timing cached reads: 14884 MB in 2.00 seconds = 7447.84 MB/sec
Timing buffered disk reads: 484 MB in 3.02 seconds = 160.02 MB/sec
```

### 4. Prosty pomiar wydajności I/O - polecenie dd

```
dd if=/dev/zero of=plik count=1 bs=1G oflag=direct
```

### 5. Weryfikacja logów systemu

Domyślnie logi zapisane są w /var/log bezpośrednio w katalogu lub z podziałem na podkatalogi z nazwą danej aplikacji.

- logi Apache: /var/log/apache tutaj mamy rozdzielenie na error.log oraz access.log
- logi PHP: /var/log/php\_error.log
- logi PostgreSQL: /var/log/postgresql/
- logi systemowe: auth - dane związane z autoryzacją  
authpriv - inne komunikaty związane z autoryzacją  
cron - komunikaty crona  
daemon - inne demony  
kern - komunikaty jądra systemu  
mail - komunikaty związane z pocztą  
syslog - komunikaty demona syslog  
user - procesy użytkowników

## 6. Skryptu uruchamiane z cron'a

Tabeli crontab wykorzystujemy do uruchamiania skryptów np do zaplanowanych zadań. Jednym z elementów eDokumentów jest Cronrunner lub backproc które dodajemy w /etc/crontab, [CronRunnera](#) lub [backproc - synchronizacja poczty w tle](#)

Skrypty CronRunnera możemy także wywoływać ręcznie z poziomu konsoli z parametrem ID procesu z tabeli crontab w bazie eDokumentów.

```
php -f /home/edokumenty/public_html/apps/edokumenty/CronRunner.php 6
```

Tabele crona są także wykorzystywane do uruchamiania skryptów np: backupowych bazy danych, plików systemu eDokumenty. Opis skryptów backupujących uruchamianych z crona znajdują się w tym [artykule](#)