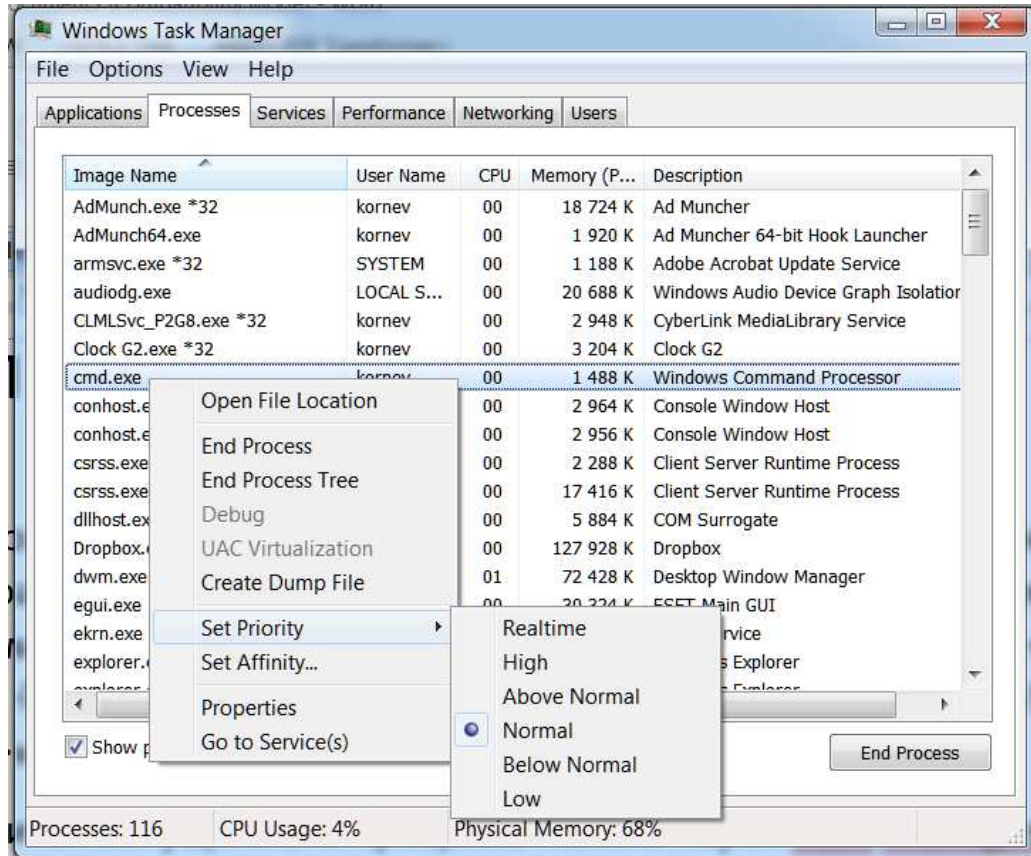


Процессы в Windows – Приоритеты

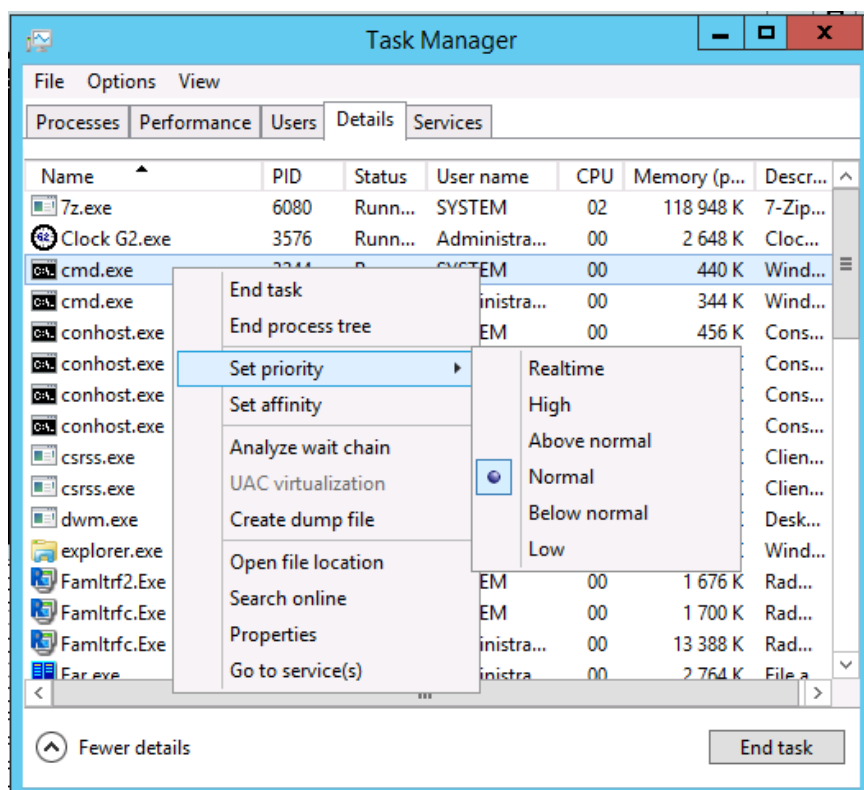
Давайте поговорим о приоритетах Windows процессов. В большинстве случаев «играться» с настройкой приоритетов нет необходимости, но, иногда, грамотный системный администратор может помочь системе более правильно распределить процессорное время между запущенными задачами. Единого рецепта нет, но путем «подбора и перебора» это вполне реализуемо. Где это может понадобиться? Например, в связке 1С-SQL можно дать больше процессорного времени 1С и SQL, как наиболее критичным к ресурсам процессам.

В общем случае, посмотреть и изменить приоритет запущенного процесса можно через Task Manager

Windows NT/2000/7/2008



В Windows 2012 это «закопали» чуть глубже



Как видно из приведенных примеров, вам доступно всего 6 приоритетов (как выяснится позже, это *классы приоритетов*). Достаточно? Microsoft считает, что да. Но давайте вспомним «легендарную» фразу Билла Гейста, который сказал, что «640 KB of RAM will be enough for everybody». Но время показало, что это далеко не так. ☺

А теперь давайте разберемся как это есть на самом деле.

На самом деле в Windows существует 32 уровня приоритета, от 0 до 31.

Они группируются так:

- 31 - 16 уровни реального времени;
- 15 - 1 динамические уровни;
- 0 - системный уровень, зарезервированный для потока обнуления страниц (zero-page thread).

При создании процесса, ему назначается один из шести *классов приоритетов*:

1. Real time class (значение 24),
2. High class (значение 13),
3. Above normal class (значение 10),
4. Normal class (значение 8),
5. Below normal class (значение 6),
6. или Idle class (значение 4).

Посмотреть приоритет процесса, как писалось выше, можно, используя Task Manager/

Примечание: Приоритеты Above normal и Below normal появились, начиная с Windows 2000.

Приоритет каждого потока (*базовый приоритет потока*) складывается из приоритета его процесса и *относительного приоритета* самого потока. Есть семь относительных приоритетов потоков:

1. Normal: такой же как и у процесса;
2. Above normal: +1 к приоритету процесса;
3. Below normal: -1;
4. Highest: +2;
5. Lowest: -2;
6. Time critical: устанавливает базовый приоритет потока для Real time класса в 31, для остальных классов в 15.
7. Idle: устанавливает базовый приоритет потока для Real time класса в 16, для остальных классов в 1.

В следующей таблице показаны приоритеты процесса, относительный и базовый приоритеты потока.

Приоритет потока	Класс процесса	Класс процесса					
		Idle class	Below normal class	Normal class	Above normal class	High class	Real time class
1		Idle	Idle	Idle	Idle	Idle	
2		Lowest					
3		Below ...					
4	Idle class	Normal	Lowest				
5		Above ...	Below ...				
6	Below normal class	Highest	Normal	Lowest			
7			Above ...	Below ...			
8	Normal class		Highest	Normal	Lowest		
9				Above ...	Below ...		
10	Above normal class			Highest	Normal		

11						Above ...	Lowest	
12						Highest	Below ...	
13	High class						Normal	
14							Above ...	
15							Highest	
15		Time critical	Time critical	Time critical	Time critical	Time critical	Time critical	
16								Idle
17								
18								
19								
20								
21								
22								Lowest
23								Below ...
24	Real time class							Normal
25								Above ...
26								Highest
27								
28								
29								
30								
31								Time critical

Теперь, когда мы все это узнали, что же с этим всем можно сделать? Ну, например, начать использовать.

Как еще можно запустить процесс с «нестандартным» приоритетом или изменить?

Метод 1. Запустить задачу/процесс и изменить приоритет через Task Manager.

Минусы метода:

- a) Доступно только 6 приоритетов
- b) Переключение приоритетов производится мышкой, не автоматизируется.

Метод 2. Можно воспользоваться командой START с соответствующими ключами

Доступные ключи, отвечающие за приоритеты, следующие (я умышленно опускаю ключи командной строки команды START не имеющие отношения к описываемому процессу работы с приоритетами):

C:\>start /?

Starts a separate window to run a specified program or command.

```
START ["title"] [/D path] [/I] [/MIN] [/MAX] [/SEPARATE | /SHARED]
  [/LOW | /NORMAL | /HIGH | /REALTIME | /ABOVENORMAL | /BELOWNORMAL]
  [/NODE <NUMA node>] [/AFFINITY <hex affinity mask>] [/WAIT] [/B]
  [command/program] [parameters]
```

```
LOW           Start application in the IDLE priority class.
NORMAL       Start application in the NORMAL priority class.
HIGH         Start application in the HIGH priority class.
REALTIME     Start application in the REALTIME priority class.
ABOVENORMAL Start application in the ABOVENORMAL priority class.
BELOWNORMAL Start application in the BELOWNORMAL priority class.
```

Как видим, команда `START` дает возможность запустить процесс все с теми же 6-ю приоритетами, которые доступны через Task Manager

Минус метода:

- a) Доступно только 6 приоритетов

Метод 3. Использование утилиты `wmic.exe`

Как было показано выше, Task Manager, и команда `START` достаточно неуклюжи для задачи назначения приоритетов. Посмотрим, как это применять более гибко. Будем использовать утилиту `wmic.exe`

Командная строка:

```
wmic process where name="AppName" CALL setpriority ProcessIDLevel
```

Пример:

```
wmic process where name="calc.exe" CALL setpriority 32768
```

или

```
wmic process where name="calc.exe" CALL setpriority "above normal"
```

Приоритеты (предопределенные):

- idle: 64
- below normal: 16384
- normal: 32
- above normal: 32768
- high priority: 128
- real time: 256

Отступление: Что делать если существует несколько одноименных процессов? Приоритет процесса можно менять как `gj` имени процесса, так и с использованием `PID` (Process ID) процесса.

Вот короткий пример запуска `wmic.exe` для получения необходимой информации

Используем команду:

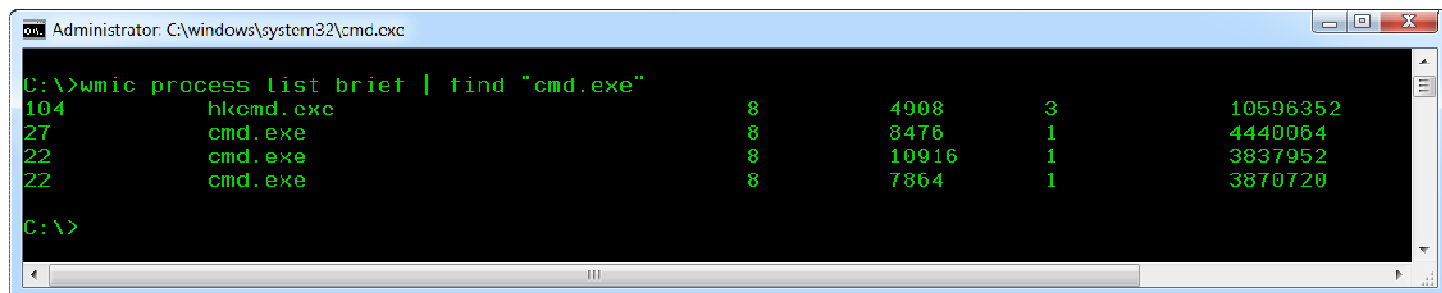
```
wmic process list brief
```

Примечание: пример выполнения этой команды приводить не буду. Слишком большой список процессов получается. Сделаете это самостоятельно, при желании.

Вы получите список процессов, запущенных на вашем локальном компьютере. Теперь выполните команду:

```
wmic process list brief | find "cmd.exe"
```

Результат:



```
Administrator: C:\windows\system32\cmd.exe
C:\>wmic process list brief | find "cmd.exe"
104      hkcmd.exe      8      4908      3      10596352
27       cmd.exe        8      8476      1      4440064
22       cmd.exe        8      10916     1      3837952
22       cmd.exe        8      7864      1      3870720
C:\>
```

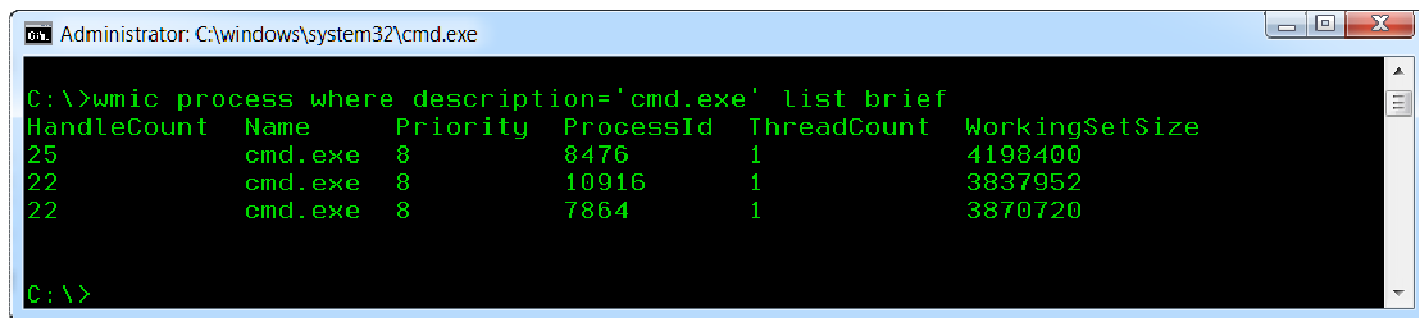
Специально запустил несколько копий `cmd.exe`, чтобы иллюстрация была более полной.

Теперь список процессов ограничен только теми процессами, в имени исполняемого модуля которых присутствует строка «`cmd.exe`». Обратите внимание на `PID` процесса(ов).

Теперь давайте попробуем отобрать интересующие нас процессы, используя непосредственно WMI и не прибегая к стандартным средствам командной строки. Для этого просто напишите:

```
wmic process where description='cmd.exe' list brief
```

Результат:



```
Administrator: C:\windows\system32\cmd.exe
C:\>wmic process where description='cmd.exe' list brief
HandleCount  Name      Priority  ProcessId  ThreadCount  WorkingSetSize
25           cmd.exe   8        8476      1            4198400
22           cmd.exe   8        10916     1            3837952
22           cmd.exe   8        7864      1            3870720
C:\>
```

Сравните полученные результаты. Запомните PID процесса CMD . EXE.

Командная строка для запуска wmic.exe

```
wmic process where processid='XXXX' CALL setpriority ProcessIDLevel
```

Ну а теперь можем изменить приоритет конкретного процесса (например с PID=8476):

```
wmic process where processid='8476' CALL setpriority 32768
```

или

```
wmic process where processid='8476' CALL setpriority "above normal"
```

А что дальше? Прикидывать, пробовать, подбирать и тонко регулировать приоритеты. Улучшая работу сервисов и процессов, а также работу конечных пользователей.