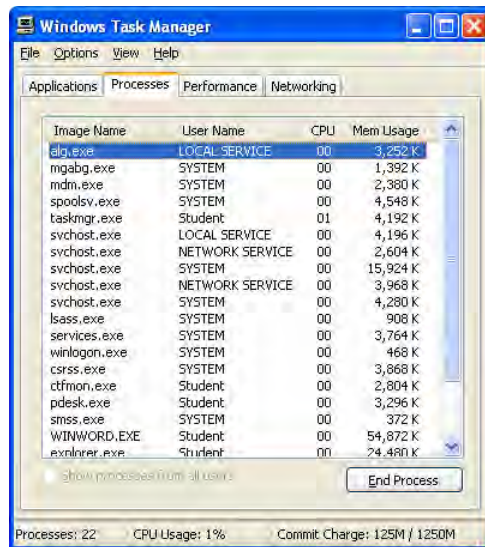


## 5

## Utjecaj prioriteta Task-ova na trajanje PC operacija

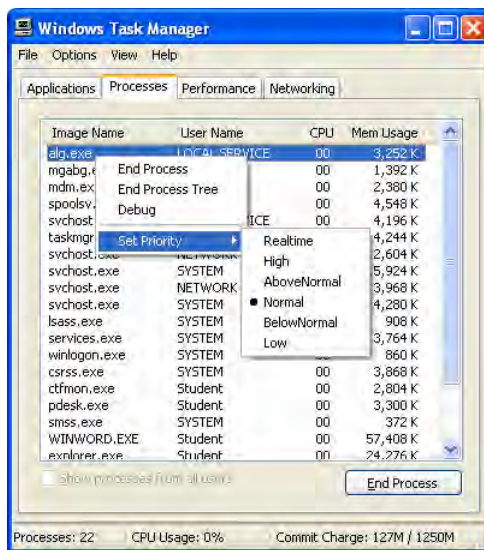
Da bi se balansirala važnost pojedinih taskova i procesorskog vremena koje im biva dodijeljeno, koriste se prioriteti izvršavanja. Za pokrenute procese moguće je vidjeti prioritet izvršavanja korištenjem *Task Manager*-a. *Task Manager* pokrećete pritiskom na *CTRL+ALT+DEL* i potom izborom „*Task Manager*“.



Slika 5-1. *Task Manager*

Desnim klikom na željeni proces i izborom „*Set Priority*“ imate mogućnost vidjeti prioritet procesa. Zavisno od vaših korisničkih privilegija, možete promijeniti prioritet izvršavanja pojedinih procesa izborom jedne od šest mogućih vrijednosti prioriteta.

## Utjecaj prioriteta Task-ova na trajanje PC operacija



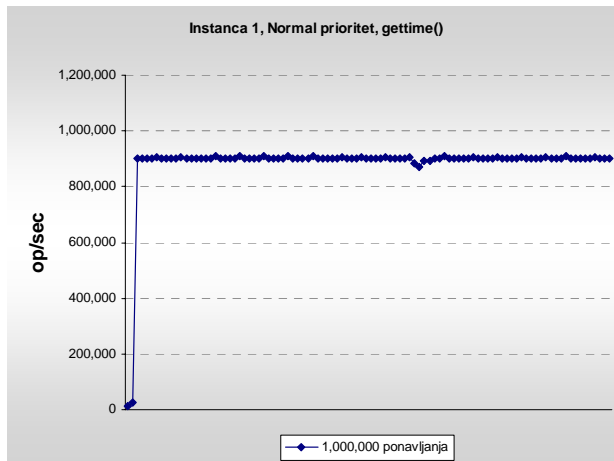
Slika 5-2. Prioritet taska

### Zadatak

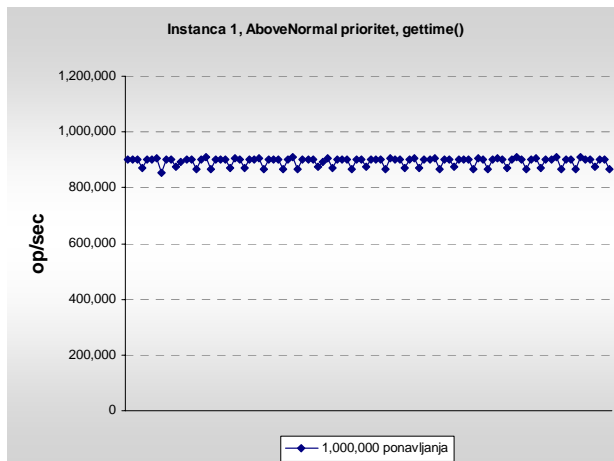
Paralelno pokrenite dvije instance aplikacije „Lab03“ koje mjere trajanje operacije `gettime`. Drugoj instanci aplikacije povećajte prioritet na „*AboveNormal*“. Rezultate mjerenja obje instance prikazite grafički.

### Rješenje

Pokrenite dvije instance aplikacije „Lab03“. Koristite *Task Manager* da promijenite prioritet instance 2 iz „*Normal*“ u „*AboveNormal*“. Obje aplikacije i oba procesa imaju identične nazive, pa možete naći da je teško povezati aplikaciju i njen proces sa prozorom aplikacije. Ovaj problem možete riješiti tako što ćete u *Task Manager*-u desnim klikom na aplikaciju i izborom „*Switch To*“ utvrditi koji prozor odgovara kojoj aplikaciji. Potom možete aplikaciju povezati sa procesom, desnim klikom na aplikaciju u „*Task Manager*-u“ i izborom „*Go To Process*“. Nakon promjene prioriteta procesa i unošenja parametara mjerenja, sve je spremno za početak mjerenja. Pokrenite instancu 1, a što prije nakon toga i instancu 2 aplikacije „Lab03“.



**Slika 5-3. Rezultati instance 1, Normal prioritet**



**Slika 5-4. Rezultati instance 2, AboveNormal prioritet**

Na prvi pogled može izgledati da se u oba slučaja gettimeofday izvršava jednako i maksimalno brzo, bez obzira na prioritet. Grafik instance 2 pokazuje čak i više odstupanja naniže od prosječne brzine gettimeofday nego što to možemo vidjeti na grafiku instance 1, gdje samo u malom području na početku grafika, postoji veliko odstupanje naniže. Ovakvo tumačenje bi sugerisalo

da se `gettime` izvršava nešto brže u tasku nižeg prioriteta, sa izuzetkom odstupanja sa početka mjerenja.

Ipak situacija je sasvim drugačija. Ne smijemo zaboraviti da grafici nisu vremenski. Ovo znači da vrijednosti koje se nalaze na istom mjestu na x osama oba grafika, se nisu obavezno desila u istom trenutku vremena. Redoslijed kojim su se događaji desili je slijedeći. Instanca 1, kao instanca višeg prioriteta, je završila svoje cjelokupno mjerenje do momenta do kojeg je instanca 1 završila svega par mjerenja. Dakle malo područje sa početka prvog grafika, u kojem se `gettime` izvršava izuzetno sporo, vremenski odgovara cjelokupnom grafiku instance 2 u kojem `gettime` juri punom brzinom kao da instanca 1 i ne postoji. U trenutku kada instanca 2 završava sa mjerenjem, instanca 1 počinje da juri punom brzinom i to je ravni dio grafika instance 1. Dakle i minimalna razlika u prioritetu između taskova, osigurava tasku većeg prioriteta neuporedivo više procesorskog vremena.