

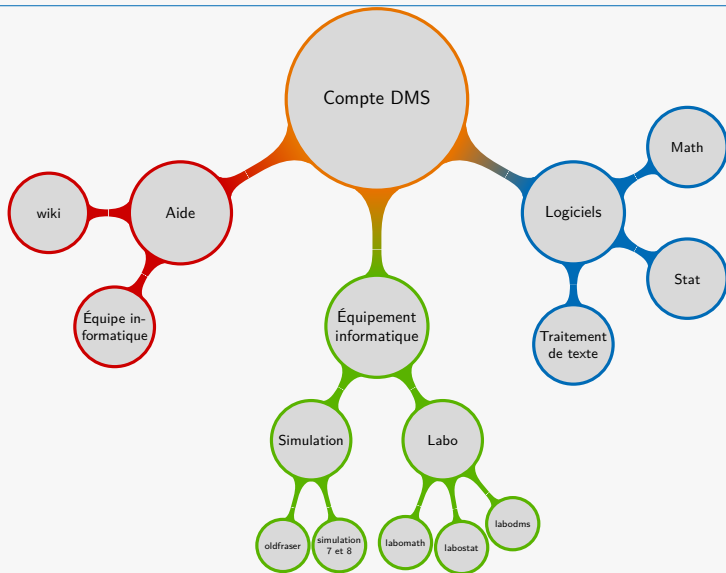
L'informatique au DMS

Comment utiliser UNIX pour faire des simulations numériques

Nicolas Bouchard

Département de mathématiques et de statistique

21 mars 2014



Le Département vous offre de la visibilité de deux façons différentes :

- ▶ Une fiche d'information est associée à votre compte. C'est à vous de la **mettre à jour** pour que cette page reflète vos intérêts de recherche.
- ▶ Vous pouvez héberger votre propre **site web** sur les serveurs du DMS.

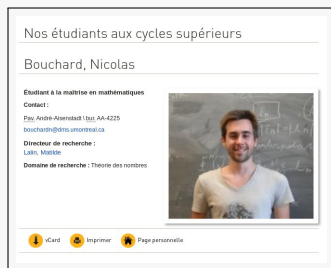
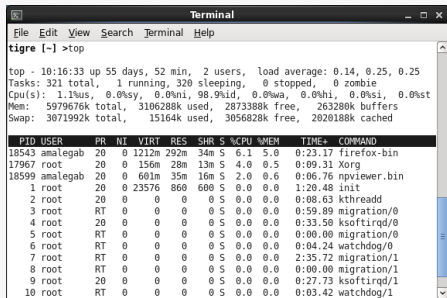


Figure : La fiche d'information de chaque étudiant est disponible en ligne sur le répertoire des étudiants.



```
Terminal
File Edit View Search Terminal Help
tigre [-] >top

top - 10:16:33 up 55 days, 52 min, 2 users, load average: 0.14, 0.25, 0.25
Tasks: 321 total, 1 running, 320 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.1%us, 0.0%sy, 0.0%ni, 98.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 5979676k total, 3106288k used, 2873388k free, 263280k buffers
Swap: 3071992k total, 15164k used, 3056828k free, 2020188k cached

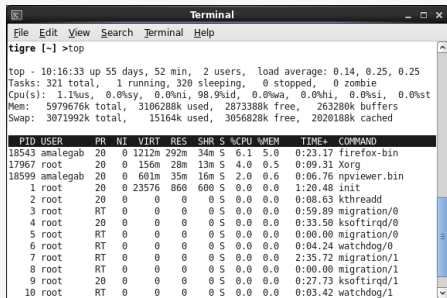
  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 18543 amalegab  20   0 1212m 292m 34m  S  6.1  5.0   0:23.17 firefox-bin
17967 root      20   0 156m  28m 13m  S  4.0  0.5   0:09.31 Xorg
18599 amalegab  20   0 601m  35m 16m  S  2.0  0.6   0:06.76 npviewer.bin
   1 root      20   0 23576  860 600  S  0.0  0.0   1:20.48 init
   2 root      20   0   0   0   0  S  0.0  0.0   0:08.63 kthreadd
   3 root      RT   0   0   0   0  S  0.0  0.0   0:59.89 migration/0
   4 root      20   0   0   0   0  S  0.0  0.0   0:33.50 ksoftirqd/0
   5 root      RT   0   0   0   0  S  0.0  0.0   0:00.00 migration/0
   6 root      RT   0   0   0   0  S  0.0  0.0   0:04.24 watchdog/0
   7 root      RT   0   0   0   0  S  0.0  0.0   2:35.72 migration/1
   8 root      RT   0   0   0   0  S  0.0  0.0   0:00.00 migration/1
   9 root      20   0   0   0   0  S  0.0  0.0   0:27.73 ksoftirqd/1
  10 root      RT   0   0   0   0  S  0.0  0.0   0:03.42 watchdog/1
```

Figure : La commande top affiche interactivement les processus en cours.

Le terminal permet un dialogue direct avec l'ordinateur. Plus concrètement, il permet par exemple de

- ▶ se connecter à d'autre machine (rapidement),
- ▶ accéder aux unités de calcul (simulation7, simulation8...),
- ▶ exécuter des programmes en arrière-plan.

Le dialogue s'effectue à l'aide de commandes.



```

tigre [-] >top

top - 10:16:33 up 55 days, 52 min, 2 users, load average: 0.14, 0.25, 0.25
Tasks: 321 total, 1 running, 320 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.1%us, 0.0%sy, 0.0%ni, 98.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 5979676k total, 3106288k used, 2873388k free, 263280k buffers
Swap: 3071992k total, 15164k used, 3056828k free, 2020188k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
18543 amalegab  20   0 1212m 292m 34m  S  6.1   5.0   0:23.17 firefox-bin
17967 root       20   0 156m  28m 13m  S  4.0   0.5   0:09.31 Xorg
18599 amalegab  20   0 601m  35m 16m  S  2.0   0.6   0:06.76 npviewer.bin
  1 root       20   0 23576  860 600  S  0.0   0.0   1:20.48 init
  2 root       20   0 0 0 0  S  0.0   0.0   0:08.63 kthreadd
  3 root      RT  0 0 0 0 0  S  0.0   0.0   0:59.89 migration/0
  4 root      20  0 0 0 0  S  0.0   0.0   0:33.50 ksoftirqd/0
  5 root      RT  0 0 0 0 0  S  0.0   0.0   0:00.00 migration/0
  6 root      RT  0 0 0 0 0  S  0.0   0.0   0:04.24 watchdog/0
  7 root      RT  0 0 0 0 0  S  0.0   0.0   2:35.72 migration/1
  8 root      RT  0 0 0 0 0  S  0.0   0.0   0:00.00 migration/1
  9 root      20  0 0 0 0  S  0.0   0.0   0:27.73 ksoftirqd/1
 10 root      RT  0 0 0 0 0  S  0.0   0.0   0:03.42 watchdog/1
    
```

Figure : La commande top affiche interactivement les processus en cours.

Quelques conventions :

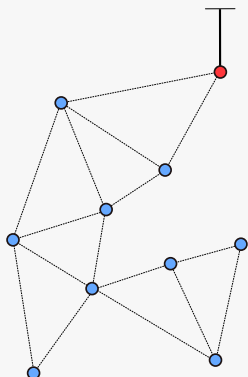
> ⇔ ce qui suit doit être tapé dans le terminal.

texte ⇔ vous devez modifier ce texte selon vos besoins.

Le terminal permet un dialogue direct avec l'ordinateur. Plus concrètement, il permet par exemple de

- ▶ se connecter à d'autre machine (rapidement),
- ▶ accéder aux unités de calcul (simulation7, simulation8...),
- ▶ exécuter des programmes en arrière-plan.

Le dialogue s'effectue à l'aide de **commandes**.



Au Département, toutes les machines sont connectées entre elles. Un seul point d'entrée est accessible depuis l'extérieur.

Se connecter au Département

- ▶ avec PuTTY (émulateur de terminal pour Windows)
- ▶ commande ssh (avec les terminaux de Mac et Linux)
- ▶ physiquement dans un laboratoire

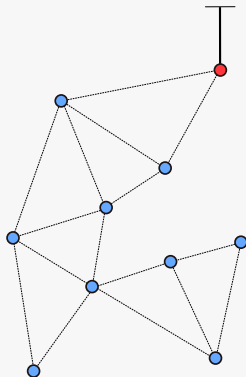
Au Département, toutes les machines sont connectées entre elles. Un seul point d'entrée est accessible depuis l'extérieur.

Se connecter au Département

- ▶ avec PuTTY (émulateur de terminal pour Windows)
- ▶ commande ssh (avec les terminaux de Mac et Linux)
- ▶ physiquement dans un laboratoire

hostname : ssh.dms.umontreal.ca

Port : 22



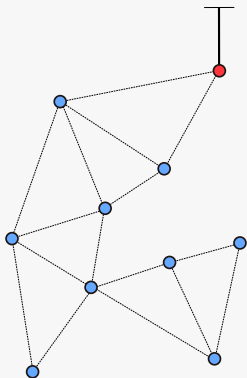
Au Département, toutes les machines sont connectées entre elles. Un seul point d'entrée est accessible depuis l'extérieur.

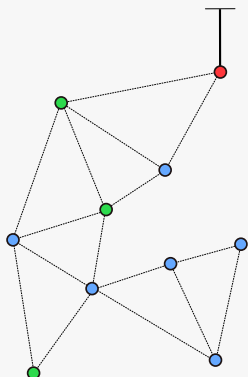
Se connecter au Département

- ▶ avec PuTTY (émulateur de terminal pour Windows)
- ▶ **commande ssh** (avec les terminaux de Mac et Linux)
- ▶ physiquement dans un laboratoire

```
> ssh usager@ssh.dms.umontreal.ca -X
```

Les utilisateur de Mac doivent télécharger XQuartz pour la redirection graphique.

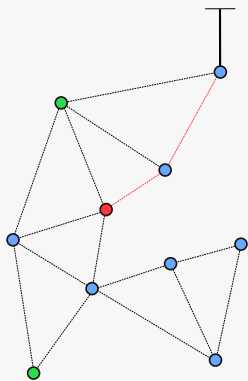




Une fois dans le réseau du Département, il faut se choisir une machine de travail. Pour accélérer les calculs, il vaut mieux choisir une machine qui n'est pas déjà trop utilisée.

Trouver les machines les moins utilisées

```
> simulation
```



Une fois dans le réseau du Département, il faut se choisir une machine de travail. Pour accélérer les calculs, il vaut mieux choisir une machine qui n'est pas déjà trop utilisée.

Trouver les machines les moins utilisées

```
> simulation
```

Une fois que l'on a identifié une machine avec une faible utilisation, on s'y connecte.

Se connecter sur une autre machine

```
> ssh nom de la machine
```

Par défaut, lorsqu'un programme est exécuté dans un terminal, l'entrée principale est le clavier et la sortie principale est la fenêtre du terminal. Heureusement, on peut modifier facilement ce comportement!

Rediriger le canal d'entrée

```
> commande < script d'entrée
```

Par défaut, lorsqu'un programme est exécuté dans un terminal, l'entrée principale est le clavier et la sortie principale est la fenêtre du terminal. Heureusement, on peut modifier facilement ce comportement!

Rediriger le canal d'entrée

```
> commande < script d'entrée
```

Rediriger le canal de sortie

```
> commande > fichier de sortie
```

Par défaut, lorsqu'un programme est exécuté dans un terminal, l'entrée principale est le clavier et la sortie principale est la fenêtre du terminal. Heureusement, on peut modifier facilement ce comportement!

Rediriger le canal d'entrée

```
> commande < script d'entrée
```

Rediriger le canal de sortie

```
> commande > fichier de sortie
```

Un exemple

```
> matlab -nodesktop < script.m > output.txt
```

Par défaut, lorsqu'un programme est exécuté dans un terminal, l'entrée principale est le clavier et la sortie principale est la fenêtre du terminal. Heureusement, on peut modifier facilement ce comportement!

Rediriger le canal d'entrée

```
> commande < script d'entrée
```

Rediriger le canal de sortie

```
> commande > fichier de sortie
```

Utiliser un pipeline (*pipelining*)

Cette procédure permet de concaténer des commandes de sorte que la sortie d'une commande est prise en entrée de la commande suivante.

```
> commande 1 | commande 2
```

Comme les calculs sont longs, on aimerait pouvoir lancer un calcul et pouvoir quitter son poste en attendant.

Lancer un processus en arrière plan

```
> programme à exécuter &
```

Vous pouvez même fermer votre session!

Comme les calculs sont longs, on aimerait pouvoir lancer un calcul et pouvoir quitter son poste en attendant.

Lancer un processus en arrière plan

```
> programme à exécuter &
```

Vous pouvez même fermer votre session! **Attention, si vous fermez votre session, il ne faut pas oublier sur quelle machine vous exécutez votre programme!**

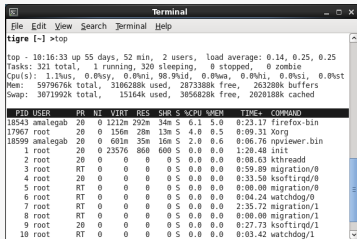
Un truc pour ne pas oublier le nom de l'ordinateur

Lorsque l'on redirige le canal de sortie, on intègre le nom de la machine dans le fichier contenant les résultats.

```
> programme à exécuter > nom de la machine_output.txt &
```


Sur une machine, il y a continuellement des programmes (processus) qui s'exécutent.

Chaque programme est identifié à l'aide d'un numéro (PID) unique. Pour effectuer une action sur un programme en cours, il faut connaître son PID. Par exemple, on peut vouloir arrêter un programme.



```
top - 10:16:33 up 55 days, 52 min, 2 users, load average: 0.14, 0.25, 0.25
Tasks: 321 total, 1 running, 320 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.1%us, 0.0%sy, 0.0%ni, 98.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 5979676k total, 3106288k used, 2873388k free, 263288k buffers
Swap: 3071992k total, 15164k used, 3056828k free, 2020188k cached
```

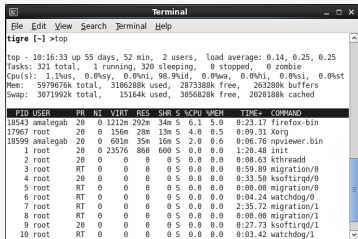
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
18543	analegab	20	0	1212m	292m	34m	S	6.1	5.0	0:23.17	firefox-bin
17967	root	20	0	156m	28m	13m	S	4.0	0.5	0:09.31	Xorg
18599	analegab	20	0	681m	35m	16m	S	2.0	0.6	0:06.76	npviewer.bin
1	root	20	0	23576	860	600	S	0.0	0.0	1:20.48	init
2	root	20	0	0	0	0	S	0.0	0.0	0:08.63	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:59.89	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:33.50	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:04.24	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	2:35.72	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
9	root	20	0	0	0	0	S	0.0	0.0	0:27.73	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:03.42	watchdog/1

Forcer l'arrêt d'un programme

```
> kill PID
```

Figure : La commande top affiche interactivement les processus en cours et permet retrouver un PID.

Sur une machine, il y a continuellement des programmes (processus) qui s'exécutent.



```
top - 10:16:33 up 55 days, 52 min, 2 users, load average: 0.14, 0.25, 0.25
Tasks: 321 total, 1 running, 320 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.1%us, 0.0%sy, 0.0%ni, 98.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 5979676k total, 3106288k used, 2873388k free, 263288k buffers
Swap: 3071992k total, 15164k used, 3056828k free, 2029188k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	MEM	TIME+	COMMAND
18543	analegab	20	0	1212m	292m	34m	S	6.1	5.0	0:23.17	firefox-bin
17967	root	20	0	156m	28m	13m	S	4.0	0.5	0:09.31	Xorg
18599	analegab	20	0	681m	35m	16m	S	2.0	0.6	0:06.76	npviewer.bin
1	root	20	0	23576	860	600	S	0.0	0.0	1:20.48	init
2	root	20	0	0	0	0	S	0.0	0.0	0:08.63	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:59.89	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:33.50	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:04.24	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	2:35.72	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
9	root	20	0	0	0	0	S	0.0	0.0	0:27.73	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:03.42	watchdog/1

Figure : La commande `top` affiche interactivement les processus en cours et permet retrouver un PID.

Chaque programme est identifié à l'aide d'un numéro (PID) unique. Pour effectuer une action sur un programme en cours, il faut connaître son PID. Par exemple, on peut vouloir arrêter un programme.

Forcer l'arrêt d'un programme

```
> kill PID
```

Forcer l'arrêt d'un programme récalcitrant

```
> kill -9 PID
```

Les comptes de tous les usagers sont soumis à des quotas stricts afin de ne pas surcharger les ressources. Un espace de 15 Go supplémentaires est disponible dans le répertoire NOBACKUP.

Accéder au répertoire NOBACKUP

```
> cd /NOBACKUP/usager
```

Ce répertoire est exclu des copies de sauvegarde du DMS. En cas de malfonction, il n'y a pas de retour possible.

Il n'y a pas de solution facile. Voici quelques pistes à explorer

- ▶ La machine est-elle déjà saturée par un autre utilisateur?

Il n'y a pas de solution facile. Voici quelques pistes à explorer

- ▶ La machine est-elle déjà saturée par un autre utilisateur?
- ▶ Y a-t-il des erreurs dans l'algorithme? Ou des optimisations possibles?

Il n'y a pas de solution facile. Voici quelques pistes à explorer

- ▶ La machine est-elle déjà saturée par un autre utilisateur?
- ▶ Y a-t-il des erreurs dans l'algorithme? Ou des optimisations possibles?
- ▶ Peut-on se passer d'écrire sur le disque dur? (Minimiser les entrées/sorties)

Il n'y a pas de solution facile. Voici quelques pistes à explorer

- ▶ La machine est-elle déjà saturée par un autre utilisateur?
- ▶ Y a-t-il des erreurs dans l'algorithme? Ou des optimisations possibles?
- ▶ Peut-on se passer d'écrire sur le disque dur? (Minimiser les entrées/sorties)
- ▶ Est-ce que les calculs sont parallélisables? (GPU, openmpi, l'extension snow de R, l'extension Jacket de Matlab...)
- ▶ Est-ce que ça vaut la peine de réécrire l'algorithme dans un langage de bas niveau?

Il n'y a pas de solution facile. Voici quelques pistes à explorer

- ▶ La machine est-elle déjà saturée par un autre utilisateur?
- ▶ Y a-t-il des erreurs dans l'algorithme? Ou des optimisations possibles?
- ▶ Peut-on se passer d'écrire sur le disque dur? (Minimiser les entrées/sorties)
- ▶ Est-ce que les calculs sont parallélisables? (GPU, openmpi, l'extension snow de R, l'extension Jacket de Matlab...)
- ▶ Est-ce que ça vaut la peine de réécrire l'algorithme dans un langage de bas niveau?

Pour les cas extrêmes, il est aussi possible de faire une demande auprès de [Calcul Québec](#) ou de [Compute Canada](#).

L'équipe informatique est à votre disposition pour répondre à vos questions. N'hésitez pas à venir nous rencontrer!

D'autres références sont disponibles dans le répertoire *Documentation* de cette présentation. Les photographies proviennent du comptes Flickr des Archives de l'Université de Montreal : [Archives/UdeM](#)

Des renseignements supplémentaires sont disponibles sur le [wiki](#) du DMS.

