cgconfig.conf

Fichier de configuration utilisé par libcgroup

Fichier de configuration utilisé par libcgroup pour définir les cgroups, leur paramètres et leur points de montage. Il consiste de sections mount, group et default.

la section mount a la forme :

```
mount {
 <controller> = <path>;
 ...
}
```

où **controller** est le nom du sous-système. Une hiérarchie nommée peut être spécifiée comme contrôleur avec "name=<somename>". **path**, le chemin où la hiérarchie cgroup associée devrait être montée.

La section group a la forme :

```
group <name> {
 [permissions]
 <controller> {
  <param name> = <param value>;
  ...
 }
...
}
```

name Nom du cgroup. le cgroup root est toujours créé automatiquement. Il peut être spécifié dans ce fichier de configuration en utilisant '.' comme nom de cgroup

permissions Les permissions du cgroup. root a toujous les permissions de faire tout avec les cgroups Ils ont la syntaxe suivante :

```
perm {
 task {
  uid = <task user>;
  gid = <task group>;
  fperm = <file permissions>
 }
 admin {
  uid = <admin name>;
  gid = <admin group>;
  dperm = <directory permissions>
  fperm = <file permissions>
 }
}
```

task user/group Nom de l'utilisateur et du group propriétaire du fichier tasks du cgroup

admin user/group Nom de l'utilisateur et du group propriétaire des autres fichiers du cgroup

controller Nom du sous-sytème kernel. Permet de spécifier le nom du fichier à définir et sa valeur

La section default a la forme :

```
default {
```

```
 perm {
  task {
   uid = <task user>;
   gid = <task group>;
   fperm = <file permissions>
  }
  admin {
   uid = <admin name>;
   gid = <admin group>;
   dperm = <directory permissions>
   fperm = <file permissions>
  }
 }
}
```

Cette section a la même forme que la section group

Exemples

Exemple 1 :

```
mount {
 cpu = /mnt/cgroups/cpu;
 cpuacct = /mnt/cgroups/cpu;
}
```

Créé la hiérarchie contrôlée par 2 sous-sytèmes. Correspond à :
mkdir /mnt/cgroups/cpu
mount -t cgroup -o cpu,cpuacct cpu /mnt/cgroups/cpu
Exemple 2 :

```
mount {
 cpu = /mnt/cgroups/cpu;
 "name=scheduler" = /mnt/cgroups/cpu;
 "name=noctrl" = /mnt/cgroups/noctrl;
}
group daemons {
 cpu {
  cpu.shares = "1000";
 }
}
group test {
 "name=noctrl" {
 }
}
```

Créé 2 hiérarchies. une 'scheduler' contrôlée par le sous-système cpu avec le group daemons en enfant. L'autre est nommée noctrl sans contrôleur, avec un groupe test. Correspond à :
mkdir /mnt/cgroups/cpu
mount -t cgroup -o cpu,name=scheduler cpu /mnt/cgroups/cpu
mount -t cgroup -o none,name=noctrl none /mnt/cgroups/noctrl
mkdir /mnt/cgroups/cpu/daemons
echo 1000 > /mnt/cgroups/cpu/daemons/www/cpu.shares
mkdir /mnt/cgroups/noctrl/tests
Exemple 3 :

```
 mount {
 cpu = /mnt/cgroups/cpu;
 cpuacct = /mnt/cgroups/cpu;
}

group daemons/www {
 perm {
  task {
   uid = root;
   gid = webmaster;
   fperm = 770;
  }
  admin {
   uid = root;
   gid = root;
   dperm = 775;
   fperm = 744;
  }
 }
 cpu {
  cpu.shares = "1000";
 }
}

group daemons/ftp {
 perm {
  task {
   uid = root;
   gid = ftpmaster;
   fperm = 774;
  }
  admin {
   uid = root;
   gid = root;
   dperm = 755;
   fperm = 700;
  }
 }
 cpu {
  cpu.shares = "500";
 }
}
```

Créer la hiérarchie contrôllée par 2 sous-système avec un group et 2 sous-groupes. Correspond à :

mkdir /mnt/cgroups/cpu
mount -t cgroup -o cpu,cpuacct cpu /mnt/cgroups/cpu
mkdir /mnt/cgroups/cpu/daemons
mkdir /mnt/cgroups/cpu/daemons/www
chown root :root /mnt/cgroups/cpu/daemons/www/*
chown root :webmaster /mnt/cgroups/cpu/daemons/www/tasks
echo 1000 > /mnt/cgroups/cpu/daemons/www/cpu.shares
mkdir /mnt/cgroups/cpu/daemons/ftp
chown root :root /mnt/cgroups/cpu/daemons/ftp/*
chown root :ftpmaster /mnt/cgroups/cpu/daemons/ftp/tasks
echo 500 > /mnt/cgroups/cpu/daemons/ftp/cpu.shares

Exemple 4 :

```
mount {
 cpu = /mnt/cgroups/cpu;
```

```
 cpuacct = /mnt/cgroups/cpuacct;
 }

group daemons {
 cpuacct{
 }
 cpu {
 }
}
```

Créé 2 hiérarchies et un group commun Correspond à :
mkdir /mnt/cgroups/cpu
mkdir /mnt/cgroups/cpuacct
mount -t cgroup -o cpu cpu /mnt/cgroups/cpu
mount -t cgroup -o cpuacct cpuacct /mnt/cgroups/cpuacct
mkdir /mnt/cgroups/cpu/daemons
mkdir /mnt/cgroups/cpuacct/daemons
Exemple 5 :

```
mount {
 cpu = /mnt/cgroups/cpu;
 cpuacct = /mnt/cgroups/cpuacct;
}

group daemons {
 cpuacct{
 }
}

group daemons/www {
 cpu {
  cpu.shares = "1000";
 }
}

group daemons/ftp {
 cpu {
  cpu.shares = "500";
 }
}
```

Créé 2 hiérarchies avec quelqes groupes dedans. un de ces groupes est créé dans les 2 hiérarchies. Correspond à :
mkdir /mnt/cgroups/cpu
mkdir /mnt/cgroups/cpuacct
mount -t cgroup -o cpu cpu /mnt/cgroups/cpu
mount -t cgroup -o cpuacct cpuacct /mnt/cgroups/cpuacct
mkdir /mnt/cgroups/cpuacct/daemons
mkdir /mnt/cgroups/cpu/daemons
mkdir /mnt/cgroups/cpu/daemons/www
echo 1000 > /mnt/cgroups/cpu/daemons/www/cpu.shares
mkdir /mnt/cgroups/cpu/daemons/ftp
echo 500 > /mnt/cgroups/cpu/daemons/ftp/cpu.shares
Exemple 6 :

```
mount {
 cpu = /mnt/cgroups/cpu;
 cpuacct = /mnt/cgroups/cpu;
}
```

```
group . {
 perm {
  task {
   uid = root;
   gid = operator;
  }
  admin {
   uid = root;
   gid = operator;
  }
 }
 cpu {
 }
}

group daemons {
 perm {
  task {
   uid = root;
   gid = daemonmaster;
  }
  admin {
   uid = root;
   gid = operator;
  }
 }
 cpu {
 }
}
```

Créé la hiérarchie contrôlée par 2 sous-sytèmes avec un groupe ayant des permissions spéciales. Correspond à :
mkdir /mnt/cgroups/cpu
mount -t cgroup -o cpu,cpuacct cpu /mnt/cgroups/cpu
chown root :operator /mnt/cgroups/cpu/*
chown root :operator /mnt/cgroups/cpu/tasks
mkdir /mnt/cgroups/cpu/daemons
chown root :operator /mnt/cgroups/cpu/daemons/*
chown root :daemonmaster /mnt/cgroups/cpu/daemons/tasks