

TD n° 6

Processus

Le but de ce TD est de vous familiariser avec les structures de données internes du noyau pour la gestion des processus et l'ordonnancement. Pour cela vous écrirez un module qui affiche la liste des processus gérés par le noyau.

1 Lister les processus en cours

1.1 Liste les processus

Ecrire un module Linux qui affiche la liste des processus en cours d'exécution lorsque le module est chargé.

La principale difficulté ici est de trouver un pointeur sur une tâche. Vous commencerez par découvrir la structure de données relative aux processus (`struct task_struct`). Puis vous tenterez de trouver un pointeur sur une tâche (n'importe laquelle, toutes les tâches étant reliés entre elles à l'aide de structures de listes doublement chaînées). Cela devrait vous permettre de trouver un point d'entrée dans la liste des processus (voir des choses qui travaillent sur la liste des processus).

Pour chaque processus, dans un premier temps, vous afficherez la commande qui lui est associée (16 premiers caractères), le PID, sa politique d'ordonnancement (NORMAL, BATCH, IDLE, FIFO ou RR) et la priorité correspondante (priorité normal ou priorité temps réel suivant le type de processus).

Dans cette première version, vous afficherez la liste des processus dans l'ordre de création, sous la forme d'une liste et terminerez par le nombre total de tâches qui fonctionnent sur votre système :

```
systemd (pid=1) NORMAL, priority=120
kthreadd (pid=2) NORMAL, priority=120
ksoftirqd/0 (pid=3) NORMAL, priority=120
kworker/0:0H (pid=5) NORMAL, priority=100
rcu_sched (pid=7) NORMAL, priority=120
rcu_bh (pid=8) NORMAL, priority=120
migration/0 (pid=9) FIFO, priority=99
watchdog/0 (pid=10) FIFO, priority=99
...
Number of tasks : 111
```

Vous prendrez soin de vérifier sur quelques processus que vous affichez bien les bonnes valeurs et donc que vous avez utilisé les bons champs dans la structure `task_struct`. Pour cela, utilisez la commande `chrt` :

Pour visualiser les informations sur un processus :

```
chrt -p PID
```

Pour afficher les valeurs min et max des priorités pour chacune des politiques et des algorithmes d'ordonnancement :

```
chrt -m
```

1.2 Arborescence des processus

Les processus sous Unix sont organisés sous une forme arborescente (processus père et processus fils). Vous modifierez le parcours entre les éléments de type `task_struct` afin de réaliser un affichage qui permette de visualiser la structure arborescente des processus. Vous utiliserez pour cela la macro `list_for_each_entry`.

Vous devriez obtenir un résultat sous la forme arborescente suivante

```
swapper/0 (pid=0) NORMAL, priority=120
  systemd (pid=1) NORMAL, priority=120
    systemd-journal (pid=181) NORMAL, priority=120
    systemd-udevd (pid=193) NORMAL, priority=120
      systemd-udevd (pid=14237) NORMAL, priority=120
```

TD n° 6

Processus

```

dhclient (pid=387)  NORMAL, priority=120
atd (pid=399)     NORMAL, priority=120
sshd (pid=400)    NORMAL, priority=120
inetd (pid=401)   NORMAL, priority=120
cron (pid=402)    NORMAL, priority=120
apt-cacher-ng (pid=404)  NORMAL, priority=120
systemd-logind (pid=408)  NORMAL, priority=120
dbus-daemon (pid=413)  NORMAL, priority=120
gpm (pid=431)     NORMAL, priority=120
irqbalance (pid=432)  NORMAL, priority=120
rsyslogd (pid=433)  NORMAL, priority=120
acpid (pid=436)   NORMAL, priority=120
login (pid=439)   NORMAL, priority=120
  bash (pid=806)   NORMAL, priority=120
    startx (pid=814)  NORMAL, priority=120
      xinit (pid=836)  NORMAL, priority=120
        Xorg (pid=837)  NORMAL, priority=119
          fluxbox (pid=846)  NORMAL, priority=120
            ssh-agent (pid=904)  NORMAL, priority=120
              xterm (pid=1151)  NORMAL, priority=120
                bash (pid=1153)  NORMAL, priority=120
                  insmod (pid=14238)  NORMAL, priority=120
                    xterm (pid=4777)  NORMAL, priority=120
                      bash (pid=4779)  NORMAL, priority=120
                        xeyes (pid=14235)  NORMAL, priority=120
                          xterm (pid=10819)  NORMAL, priority=120
                            bash (pid=10821)  NORMAL, priority=120
exim4 (pid=771)   NORMAL, priority=120
VBoxService (pid=790)  NORMAL, priority=120
systemd (pid=803)  NORMAL, priority=120
  (sd-pam) (pid=804)  NORMAL, priority=120
VBoxClient (pid=868)  NORMAL, priority=120
  VBoxClient (pid=870)  NORMAL, priority=120
kthreadd (pid=2)  NORMAL, priority=120
ksoftirqd/0 (pid=3)  NORMAL, priority=120
kworker/0:0H (pid=5)  NORMAL, priority=100
rcu_sched (pid=7)   NORMAL, priority=120
rcu_bh (pid=8)     NORMAL, priority=120
migration/0 (pid=9)  FIFO, priority=99
watchdog/0 (pid=10)  FIFO, priority=99
watchdog/1 (pid=11)  FIFO, priority=99
migration/1 (pid=12)  FIFO, priority=99
ksoftirqd/1 (pid=13)  NORMAL, priority=120
kworker/1:0H (pid=15)  NORMAL, priority=100

```

Vous prendrez soin de vérifier que le résultat correspond bien à l'arborescence que vous obtenez avec la commande utilisable dans l'espace utilisateur suivante :

```
pstree
```

Quelle différence majeure faites-vous entre le résultat que vous obtenez et le résultat de la commande `pstree` ?

Que notez-vous sur les trois premiers processus créés ? Quels sont-ils à quoi correspondent-ils ? Les processus « dans l'espace utilisateur » sont regroupés sous quel processus père.

TD n° 6

Processus

2 Consultation d'informations pertinentes sur l'ordonnanceur

Vous pouvez consulter, depuis l'espace utilisateur, des informations complémentaires sur l'ordonnancement des processus sur votre machine (en particulier la répartition des processus sur vos multiples cœurs). Les fichiers suivants peuvent être consultés : `/proc/sched_debug`, `/proc/schedstat` et `/proc/PID/sched`.

```
cat /proc/sched_debug  
cat /proc/schedstat  
cat /proc/`pidof cmd`/sched
```

3 Amélioration du module

Vous modifierez votre module afin de fournir ces informations pertinentes via un fichier dans `/proc` et non plus simplement au chargement du module.