

Agendamento de Tarefas com cron e at

Programar a execução de scripts ou comandos em **horários específicos** ou em **intervalos tempo regulares**, no Linux podemos usar os serviços **cron** ou **at**, no qual cada usuário (roots ou outros usuários do sistema) possui sua própria agenda. O serviço **cron** funciona verificando a hora e posteriormente se há ou não algum comando a ser executado. Esse serviço é mais indicado para agendar tarefas que devem ser repetidas diversas vezes. No subsistema linux para windows (WSL) apenas o usuário root pode agendar tarefas usando o **cron** e o **at**.

1. Serviço e Comando cron

Colocar os scripts a serem rodados em uma destas pasta:

/etc/cron.daily (a cada dia)
/etc/cron.hourly (a cada hora)
/etc/cron.weekly (a cada semana)
/etc/cron.monthly (a cada mês)

Locais Importantes para o Serviço cron

/etc/crontab
/etc/cron.d/ # cron verifica alterações a cada dia nessa pasta.
/var/spool/cron.a # cron verifica alterações a cada minuto nessa pasta.
/var/spool/cron # arquivos cron de agendamento

Comando crontab

\$ crontab -e # abrir arquivo para editar os agendamentos criar ou editar
\$ crontab -l # listar agendamentos feitos pelo usuário (mostrar arquivo crontab)
\$ crontab -r # remover agendamentos ou arquivos de crontab
\$ crontab -v # mostrar quando foi a última edição dos agendamentos/crontab

Definir e-mail destinatário para o crontab de cada usuário

\$ crontab -e
Alterar endereço de e-mail na seguinte linha do arquivo
MAILTO= endereço de e-mail

O usuário pode alterar os agendamento/ crontab dos outros usuários

Configuração do arquivo de agendamento/crontab

	[minuto]	[hora]	[dia do mês]	[mês]	[dia da semana]	[usuário]	[local/script]
Opções	0-59	0-23	1-31	1-12	0-6	opcional	

Obs: Colocar sempre espaço entre os campos (minutos, hora, mês e etc)

Representação de Intervalos tempo:

2,7,9 (Nas horas, segundos ou dias 2, 7 e 9)

10-15 (De 10 até 15)

* (todos os valores válidos naquele campo)

Dias da semana: 0 (Domingo), 1(Segunda), 2(Terça), 3(Quarta), 4 (Quinta), 5 (Sexta) e 6 (Sábado).

* /5 (a cada 5 minutos, segundos, dias, meses e etc)

Horas: 12-16/2 (A cada 2 horas entre 12:00 até as 16:00)

Exemplos:

Executar script de backup diariamente às 8:00, 12:00 e 16:00

\$ crontab -e

Adicionar no arquivo de agendamento

0 8,12,16, * * * /home/alisandra/backup.sh

Executar script de backup às 18:30 a cada três meses

\$ crontab -e

Adicionar no arquivo de agendamento

30 18 * */3 * /home/alisandra/backup.sh

Executar script de backup às 9:15 no dia 1º de janeiro á março

\$ crontab -e

Adicionar no arquivo de agendamento

15 9 1 1-3 * /home/alisandra/backup.sh

Executar script de backup 2 vezes nos dias de segunda-feira e quinta-feira:

\$ crontab -e

Adicionar no arquivo de agendamento

0 */12 * * 1,4 /home/alisandra/backup.sh # Realizar a cada 12 horas

0 8,18 * * 1,4 /home/alisandra/backup.sh # Ou definindo hora específica

Executar script de backup 1 vez por semana

\$ crontab -e

Adicionar no arquivo de agendamento

0 0 * * 1 /home/alisandra/backup.sh # executar sempre na segunda as 00:00

Executar script de backup 4 vezes por dia durante a semana, exceto no final de semana

0 */6 * * 1-5 /home/alisandra/backup.sh # a cada 6 horas o script será executado

0 8,12,16,22 * * 1-5 /home/alisandra/backup.sh # Ou especificando a hora

Depois de salvar alterações no arquivo de agendamento **crontab**, lembrar de reiniciar o **cron** para que o novo agendamento seja executado.

\$ /etc/init.d/cron restart

2. Serviço e Comando at

Permite agendar uma tarefa **uma vez** semelhante ao cron

\$ at quando # escrever comandos e pressionar **ctrl+d**

\$ at -f comando quando

\$ comando | at quando

Opções de quando:

now

3am + 2 days

12:30 abril 22 (22 de abril às 12:30)

now + 10 minutes

midnight

6pm today

6am tomorrow

Ver agendamentos do programados:

\$ at -l

Exemplos:

Execute script.sh daqui a 5 minutos

\$ at -f script.sh now+5minutes

Execute script.sh dia 1 de maio as 16h

\$ at -f script.sh 16:00 may 1

Execute o comando echo amanhã as 8:00 da manhã

\$ echo | ls /var/log | at 8am tomorrow

Ver agendamentos no at

\$ ls -l /var/spool/at

ou

\$ ls -l /var/spool/cron/atjobs

ou

\$ at -l

Remover agendamentos do at

\$ atrm número-agendamento (ver numero com: \$ at -l)

O uso do comando **at** pode ser gerenciado através de listas de usuários que podem (**/etc/at.allow**) ou não podem (**/etc/at.deny**) realizar agendamento. Caso não existam esses arquivos todos os usuários serão capazes de realizar agendamentos.