Sistemas Operacionais 2

shell

Hélio Crestana Guardia
helio@dc.ufscar.br

Universidade Federal de São Carlos
Departamento de Computação

Shell: interpretador de comandos

- Ex: *bash, sh, csh, tcsh, ash, ksz, zsh, ...(/etc/shells)*
- Ativação:
  - *Login* em modo texto (*login shell*)
  - Janela no modo gráfico (*interactive shell*), iniciado por linha de comando
- Configurações:
  - Variáveis de ambiente: PATH, HOME, USER, ...
    - *PATH=/bin:/usr/bin:/usr/lib/java/bin:~/bin*
    - *export PATH=$PATH:.*
  - ***Aliases***: apelidos para comandos
    - *alias rm='rm -i'; alias cp='cp -i'*
  - Configuração do terminal: ***stty***
    - *stty -a; stty erase ^h*
  - Histórico de comandos
    - HISTSIZE, HISTFILE, HISTFILESIZE
- Arquivos de configuração (*bash*):
  - *Login shell*: /etc/profile, (/etc/profile.d/*), ~/.bash_profile, ~/.bash_login e ~/.profile
  - *Interactive shell*: ~/.bashrc
- Fim da sessão:
  - *Login shell*: *logout*
  - *Interactive shell*: exit, <ctrl>d // EOF

Gerenciamento de processos

- Processo = programa em execução

[shell]# *prog [parâmetros] [redirecionamentos] [&] <enter>*

- Identificadores: *pid*, *ppid*, *gid*, *uid, pgid*
- Aspectos:
  - Parâmetros: *int main(int argc, char **argv)*
  - Entrada e saída: *stdin, stdout, stderr*
  - Política de escalonamento e prioridades
  - Tratamento de sinais
  - Limites
  - Monitoração do consumo de recursos
- Término:
  - Voluntário: exit(), return, _exit()
  - Recebimento de **sinal** (*trap* ou externo)

Controle manual de processos

Ativação e redirecionamento de entrada e saída:
 # prog // execução em primeiro plano: *foreground*
 # prog & // execução em segundo plano: *background*
 # prog > saída // *redirecionamento de stdout*
 # prog >> saída // *redirecionamento de stdout (append dados no fim do arquivo)*
 # prog 2> erros // *redirecionamento de stderr*
 # prog 2>> erros// redirecionamento de stderr (append dados no fim do arquivo)
 # prog <entrada // redirecionamento de *stdin*

Redirecionamento de E/S com ***pipe***:
prog1 | prog 2 // cria pipe, redireciona *stdout* de prog1 para *pipe* e *stdin* de prog2 para *pipe*

Programas iniciados a partir do *shell* corrente:
jobs
fg [%job]
bg [%job]

Tratamento do SIGHUP:
 # nohup prog ... & // permite terminar *shell* sem que proc em *background* seja terminado

Controle do nível de ***nice*** (prioridade):
 # nice -n X prog [args] // inicia prog com nível de *nice* igual a X
 # renice X pid // altera nível de *nice* de processo já em execução

Controle manual de processos

Identificação dos processos e seus ***pids***:

 # ps // lista processos iniciados a partir do *shell* corrente
 # ps -f // lista processos com informações sobre seus identificadores
*# ps -U _login_* // lista processos do usuário _login_
 # man ps // !

Sinais podem ser enviados aos processos pelo SO, indicando eventos ocorridos em suas execuções, ou por outros processos, para comunicação ou sincronização.

Envio de sinais para o processo em *foreground* no *shell* corrente:

 # stty -a // lista configuração do terminal, incluindo combinações de teclas que geram o
// envio de sinal ao processo corrente (em *foreground*)
<ctrl>c // interrompe processo – envia SIGINT
<ctrl>z // pára processo – envia SIGTSTP
intr CHAR: will send an interrupt signal
susp CHAR: will send a terminal stop signal
dsusp CHAR: will send a terminal stop signal once input flushed
quit CHAR: will send a quit signal
stop CHAR: will stop the output
start CHAR: will restart the output after stopping it

Controle manual de processos

Envio de sinais para processo em função do ***pid***:

 # kill pid // por *default*, envia sinal SIGTERM (15), cuja ação padrão é terminar o processo
 # kill -_SIGxxx_ pid // envia sinal SIGxxx

Envio de sinais para processo pelo nome:

 # killall nome_proc // por *default*, envia sinal SIGTERM (15)
 # killall -_SIGxxx_ pid // envia sinal SIGxxx

Envio de sinais para processo por outros parâmetros:

 # pkill [-SIGNAL] [-fvx] [-n|-o] [-P PPIDLIST] [-g PGRPLIST] [-s SIDLIST]
[-u EUIDLIST] [-U UIDLIST] [-G GIDLIST] [-t TERMLIST] [PATTERN]