

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيمِ



دانشگاه صنعتی امیرکبیر  
(پل تکنیک تهران)



دانشکده مهندسی  
کامپیوتر و فناوری اطلاعات

# کارگاه کامپیووتر

## Computer Workshop

ارائه دهنده: مهدی خادمیان

نیمسال اول ۹۳-۹۴



# عنایین مطالب جلسه قبل ...

- آشنایی با Linux Command Shell
- آشنایی با ساختار پوشه‌بندی Linux
- پوشه جاری، پوشه خانه، پوشه بالاتر
- آشنایی با چند دستور پرکاربرد نظیر:
  - ls, cd, mkdir, rmdir, cp, mv, rm, pwd
  - cat, man, echo
- آشنایی با فایل‌های مربوط به اطلاعات پردازنده و حافظه
- /proc/cpuinfo, /proc/meminfo
- مشاهده تاریخچه دستورات

# ... عنایین مطالب جلسه قبل

- استفاده ابتدایی از ویرایشگر محبوب **! vim**
- تغییر خروجی استاندارد اجرای برنامه‌ها توسط < و >>
- اجرای برنامه‌ها در پس زمینه توسط &
- مدیریت **process**‌ها توسط:
  - **top**
  - ؟
  - **killall**
  - **kill**
  - **ps**

# مطالبی که بیشتر توضیح می‌دهیم

- تغییر کاربر جاری توسط `sudo` و `SU`
- تغییر سطح دسترسی به فایل‌ها توسط `chmod`
- `mount` نمودن - سوار کردن
- فشرده‌سازی فایل‌ها توسط `tar`

# مرور چند مطلب

## دانشکده فنی و فنون اسلامی

# تغییر کاربر جاری

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

- کاربر جاری:

- **whoami**

- تغییر کاربر جاری:

- **su (substitute user)**

- **su root, su alice, su user**

# اجرا توسط کاربر سطح بالا؟!

- **sudo** (hmm... what's the mnemonic? Super-User-DO?)
- **sudo vim /etc/profile**

# اجرا توسط کاربر سطح بالا؟!

- **sudo** (hmm... what's the mnemonic? Super-User-DO?) is completely different. It uses a config file (`/etc/sudoers`) which lists which users have rights to specific actions (run commands as root, etc.) When invoked, it asks for the password of the user who started it - to ensure the person at the terminal is really the same "joe" who's listed in `/etc/sudoers`. To revoke admin privileges from a person, you just need to edit the config file (or remove the user from a group which is listed in that config). This results in much cleaner management of privileges.
- As a result of this, in Debian-based systems root user has no password set - i.e. it's not possible to login as root directly.
- Also, `/etc/sudoers` allows to specify some additional options - i.e. user X is only able to run program Y etc.





# مطلوب این جلسه

- تعیین مفسر command shell
- مشاهده، استفاده و تغییر Environment Variables
- خواندن ورودی از کاربر در برنامه shell
- تحويل ورودی به برنامه توسط <
- برنامه نویسی bash

# تعیین مفسر command shell

- مفسر پیش فرض ArchLinux

- zsh

- مفسرهای پیش فرض اغلب توزیع‌های Linux

- sh, bash

- تعیین مفسر توسط دنباله علامت‌های #! - مثال:

- #!/bin/sh
- #!/bin/bash
- #!/bin/zsh

- مشاهده لیست مفسرهای سیستم:

- cat /etc/shells

# Environment Variables

## Environment Variables

- مشاهده لیست
- استفاده از یک متغیر:
- **\$SHELL, \$HOME, \$PATH**
  - در windows
- **%PATH%, %TMP%, %WinDir%**
  - تغییر :Environment Variables
  - ویرایش فایل‌های مربوطه توسط کاربر root
  - **/etc/bash.bashrc, /etc/environment, /etc/profile**

# خواندن ورودی از کاربر

- خواندن ورودی توسط دستور زیر انجام می‌شود:
- **read [options] NAME1 NAME2 ... NAMEN**
- مثال:
- **echo "Please enter some input: "**
- **read input\_variable**
- **echo "You entered: \$input\_variable"**

# برنامه نویسی

## بازیگر

# تعريف متغير و commenting

- **var=value** #Assign the string value to shell variable var.
- **\$var** #Value of shell variable var.
- **\${var}abc** #Example: value of shell variable var with string abc appended.
- **#** #At start of line, indicates a comment.

# مقایسه‌های عددی

- **n1 -eq n2**      **#Check to see if n1 equals n2.**
- **n1 -ne n2**      **#Check to see if n1 is not equal to n2.**
- **n1 -lt n2**      **#Check to see if n1 < n2.**
- **n1 -le n2**      **#Check to see if n1 <= n2.**
- **n1 -gt n2**      **#Check to see if n1 > n2.**
- **n1 -ge n2**      **#Check to see if n1 >= n2.**

# مقایسه مایین رشته‌ها

- **s1 = s2** #Check if s1 equals s2.
- **s1 != s2** #Check if s1 is not equal to s2.
- **-z s1** #Check if s1 has size 0.
- **-n s1** #Check if s2 has nonzero size.
- **s1** #Check if s1 is not the empty string.

• مثال:

```
• if [ $myvar = "hello" ] ; then  
• echo "We have a match"  
• fi
```

# بلوک if

- **if [boolean statement]**
- **then**
- **echo "true"**
- **else**
- **echo "false"**
- **fi**

- **for arg in [list]**
- **do**
- **echo \$arg**
- **done**
  
- **for arg in [list] ; do**
- **echo \$arg**
- **done**

# مثال: ...

... . . .

- **NUMBERS="1 2 3"**
- **for number in `echo \$NUMBERS`**
- **do**
- **echo \$number**
- **done**
  
- **for number in \$NUMBERS**
- **do**
- **echo -n \$number**
- **done**
  
- **for number in 1 2 3**
- **do**
- **echo -n \$number**
- **done**

# بلوک while

- **while [ expression ]**
- **do**
- **commands**
- **done**
  
- **until [ expression ]**
- **do**
- **commands**
- **done**

# بلوک switch, case

- **case string1 in  
str1)  
commands;;  
str2)  
commands;;  
\*)  
commands;;  
esac**

# اجراهای مشروط

- **cmd1 && cmd2** #Run cmd1, then if cmd1 successful run cmd2, otherwise skip.
- **cmd1 || cmd2** #Run cmd1, then if cmd1 not successful run cmd2, otherwise skip.
- **cmd1; cmd2** #Do cmd1 and then cmd2.
- **cmd1 & cmd2** #Do cmd1, start cmd2 without waiting for cmd1 to finish.

