



ORVIETOLINUX APS PRESENTA:

CORSO LINUX

GESTIONE DI UN SERVER

LEZIONE 2

**COMPILAZIONE DI SOFTWARE OPENSOURCE
BASI DI BASH SCRIPTING, WEBSERVICE L.A.M.P.**



PIACERE DI (RI)CONOSCERVI

ORVIETO LINUX USER GROUP

ASSOCIAZIONE DI PROMOZIONE SOCIALE

OBIETTIVO DIFFONDERE CULTURA

SOFTWARE LIBERO ED OPEN SOURCE

MARCO CIAMMELLA

PRESIDENTE ORVIETOLINUX APS

CUSTOMER EVOLUTION SPECIALIST



OBIETTIVI DELLA LEZIONE #2

SAPER COMPILARE UN SOFTWARE DA SORGENTI

**REALIZZARE PICCOLI SCRIPT IN BASH
PER AUTOMATIZZARE IL SISTEMA**

**INSTALLARE UN SERVIZIO APACHE+MYSQL+PHP
PER OSPITARE PAGINE WEB**



COMPILAZIONE DA SORGENTI

```
checking for dbus-binding-tool... yes
checking for DBUS... yes
checking for NETWORKMANAGER... yes
checking for python... /usr/bin/python
checking location of the D-Bus services directory... /usr/share/dbus-1/services
Building with D-Bus support
checking for Python compile flags... ok
checking for perl... /usr/bin/perl
checking for Perl compile flags... ok
checking for libperl... checking for perl_run... yes
checking EXTERN.h usability... yes
checking EXTERN.h presence... yes
checking for EXTERN.h... yes
checking for perl.h... yes
checking for perl... perl
checking for perl module ExtUtils::MakeMaker... ok
checking for Perl_eval_pv... no
checking for old perl... no
checking for DynaLoader.a... no
checking for GnuTLS includes... ""
checking gnutls/gnutls.h usability... yes
checking gnutls/gnutls.h presence... yes
checking for gnutls/gnutls.h... yes
checking for GnuTLS libraries...
```



PARTIAMO DALLE DEFINIZIONI

OPEN SOURCE = CODICE SORGENTE APERTO

CODICE SORGENTE (?)

PROCEDURA: LISTATO > CONFIGURAZIONE > COMPILAZIONE

RISULTATO: FILE BINARIO ESEGUIBILE



STRUMENTI UTILI ALLA COMPILAZIONE

`sudo apt install build-essential`



ALGORITMI DI COMPRESSIONE - RIPASSO

bzip2 file //COMPRESSIONE

bzip2 -d file.bz2 //DECOMPRESSIONE

tar cjf file.tar.bz2 //LISTA FILE

tar xjf file.tar.bz2 //ESTRAZIONE



ESERCIZIO 0

- 1) SCARICARE nmap DA <https://nmap.org/download.html>
- 2) ESTRARRE IL CONTENUTO DA TERMINALE CON bzip2
- 3) VERIFICARE CHE IL COMPILATORE SIA INSTALLATO
- 4) COMPILARE I SORGENTI
- 5) LANCIARE IL COMANDO nmap



ESERCIZIO 0 - SVOLGIMENTO

1) `wget https://nmap.org/dist/nmap-7.91.tar.bz2`

2) `tar xjf nmap-7.91.tar.bz2`

3) `dpkg -s build-essential`

4) `cd nmap-7.91/`

`./configure`

`make`

`sudo make install`

5) `nmap -sV localhost -A`



BASH SCRIPTING

BASH = bourne again shell

Scripting Bash:

**linguaggio di programmazione interpretato
con il BASH si realizzano script**

**i programmi destinati
all'automazione del sistema operativo
si definiscono batch**



BASH SCRIPTING

ESEMPIO SINTASSI:

#!/bin/bash → DEFINISCE LA SHELL DI RIFERIMENTO

echo "messaggio" → STAMPA A VIDEO UNA STRINGA

exit → CHIUDE LO SCRIPT

ESEMPIO COMANDO DI ESECUZIONE:

./nomescript.sh

→ ESEGUITO NELLA WORK DIRECTORY

IN CUI RISIEME LO SCRIPT BASH



ESERCIZIO 1

**REALIZZARE UNO SCRIPT IN BASH CHE PERMETTA DI
VISUALIZZARE A SCHERMO LA SCRITTA:
LINUX FOREVER**



ESERCIZIO 1 - SVOLGIMENTO

```
pico script_1.sh  
#!/bin/bash  
echo "Linux Forever"  
exit
```

ESECUZIONE DA RIGA DI COMANDO:

```
$ ./script_1.sh
```

FUNZIONA?



ESERCIZIO 1 - SVOLGIMENTO

```
chmod +x script_1.sh
```

```
./script_1.sh
```

ORA FUNZIONA!



BASH SCRIPTING - VARIABILI

LE VARIABILI:

**dati (sequenze di caratteri e/o cifre)
modificabili in una porzione di memoria**

ES.

data=20200307

pigreco=3.14

nome=marco

toponomastica=via



ESERCIZIO 2

**CREARE UNO SCRIPT BASH
CHE STAMPI A VIDEO UN MESSAGGIO CON UN NOME,
AD ESEMPIO "CIAO LINUX!",
UTILIZZANDO UNA VARIABILE**



ESERCIZIO 2 - SVOLGIMENTO

```
$ pico script_2.sh  
#!/bin/bash  
# script n.2  
nome=Linux  
echo "Ciao"  
echo $nome  
exit
```

```
$ chmod +x script_2.sh && ./script_2.sh
```



BASH SCRIPTING – USO DI ARGOMENTI

ESEMPIO DI SINTASSI:

```
./script.sh <argomento1> <arg2> ... <argN>
```

```
#!/bin/bash
```

```
echo "$0 $1 $N"
```



ESERCIZIO 3

**CREARE UNO SCRIPT CHE PRENDA COME ARGOMENTO
IL TUO NOME E COGNOME E LO STAMPI A VIDEO**



ESERCIZIO 3 - SVOLGIMENTO

`./script_3.sh nome cognome`

`#!/bin/bash`

`echo "Mi chiamo $1 $2"`



BASH SCRIPTING – COMANDI DI SISTEMA

**è possibile richiamare comandi come ls, chmod, ...
all'interno dello script**

ES.

#!/bin/bash

<comando> <argomento>



BASH SCRIPTING – COMANDI DI SISTEMA

VARIABILI DI AMBIENTE:

\$ env

ESEMPIO:

PWD → /home/utente/

USERNAME → utente

LANG → lingua di sistema



ESERCIZIO 4

**CREARE UNO SCRIPT CHE VISUALIZZI IL CONTENUTO
DELLA DIRECTORY CORRENTE, ANCHE CON L'USO DELLE
VARIABILI DI SISTEMA**



ESERCIZIO 4 - SVOLGIMENTO

TIPOLOGIA A: script_4a.sh

#!/bin/bash

ls -la

TIPOLOGIA B: script_4b.sh

#!/bin/bash

ls \$PWD



AUTOMATIZZARE IL SISTEMA



ESERCIZIO 5 – CREIAMO UN SERVIZIO CUSTOM

**CONFIGURARE IL BOOT DI UBUNTU SERVER
IN MODO CHE VENGA CALCOLATA L'OCCUPAZIONE DEL DISCO
AD OGNI AVVIO TRAMITE SCRIPT IN BASH**



ESERCIZIO 5 - SVOLGIMENTO

```
$ sudo pico /usr/local/bin/occupazione.sh
```

```
#!/bin/bash
```

```
date > /root/occupazione.txt
```

```
df -h >> /root/occupazione.txt
```



ESERCIZIO 5 - SVOLGIMENTO

```
$ sudo pico /etc/systemd/system/occupazione.service
```

```
[Unit]
```

```
After=network.target
```

```
[Service]
```

```
ExecStart=/usr/local/bin/occupazione.sh
```

```
[Install]
```

```
WantedBy=default.target
```



ESERCIZIO 5 - SVOLGIMENTO

```
sudo chmod 664 /etc/systemd/system/occupazione.service
```

```
sudo chmod 744 /usr/local/bin/occupazione.sh
```

```
systemctl daemon-reload
```

```
systemctl enable occupazione.service
```

```
systemctl status occupazione.service
```

```
sudo reboot
```

```
sudo ls /root/
```



USO DEI CRON

UN CRON CONSENTE DI DEFINIRE
QUANDO AVVIARE DEI COMANDI DI SISTEMA IN DETERMINATI
MOMENTI DI UNA GIORNATA / SETTIMANA / MESE / ANNO

EDITOR / SCHEDULER DEI CRON: `sudo crontab -e`

SINTASSI: `mm hh GG MM G(W) <comando>`

ESEMPIO: `15 13 * * * /usr/local/bin/occupazione.sh` // * = qualsiasi



ESERCIZIO 6

**IMPOSTARE UN CRON PER
MANTENERE SEMPRE AGGIORNATI
I REPOSITORY DI UBUNTU**



ESERCIZIO 6 - SVOLGIMENTO

```
sudo crontab -u root -e  
15 16 * * * /usr/bin/apt update
```

COME TROVO IL PERCORSO DEI COMANDI?

which nome-comando



ESERCIZIO 7

PROGRAMMARE LO SPEGNIMENTO DEL SERVER



ESERCIZIO 7 - SVOLGIMENTO

```
sudo crontab -u root -e  
05 16 * * * /sbin/shutdown -h now
```



ESERCIZIO 8

PROGRAMMARE IL RIAVVIO DEL SERVER



ESERCIZIO 8 - SVOLGIMENTO

```
sudo crontab -u root -e  
10 16 * * * /sbin/reboot
```



SERVER L.A.M.P.

LINUX – APACHE – MYSQL – PHP





Apache

**SISTEMA INFORMATICO
NELLO SPECIFICO È UN WEB SERVER HTTP**

**FORNISCE RISORSE TRAMITE IL PROTOCOLLO HTTP
HYPERTEXT TRANSFER PROTOCOL**

**RISORSE = PAGINE WEB (HTML O ALTRO LINGUAGGIO)
AD UN CLIENT HTTP CHE NE FA RICHIESTA SULLA PORTA 80**



RDBMS - GESTORE DI DATABASE RELAZIONALI
MYSQL GESTISCE PIÙ DATABASE INSIEME

DATABASE È UN SISTEMA CHE
MEMORIZZA INFORMAZIONI DI DIVERSO TIPO
SOTTOFORMA DI RECORD ORGANIZZATI IN TABELLE

PORTA 3306



- LIVELLO SISTEMA: INTERPRETE
- LIVELLO APPLICATIVO: LINGUAGGIO DI PROGRAMMAZIONE

PHP HYPERTEXT PREPROCESSOR
SVILUPPO APPLICAZIONI WEB LATO SERVER
PAGINE DINAMICHE



RIPASSO SULLA GESTIONE DEI SERVIZI

systemctl

SINTASSI:

systemctl <start | stop | restart | status> <servizio>

ESEMPIO:

systemctl start httpd
systemctl stop mysqld
systemctl restart sshd



INSTALLARE I WEB SERVICE

```
sudo apt update  
sudo apt install lamp-server^
```



ATTIVARE IL WEB SERVER APACHE

```
sudo service apache2 start  
sudo systemctl status apache2
```



ATTIVARE IL DBMS MYSQL

```
sudo service mysql start  
mysqld - -version
```

```
sudo mysql_secure_installation //messa in sicurezza
```



ALCUNI COMANDI UTILI PER MYSQL

sudo mysql

mysql > show databases;

mysql > create database nome-db;

mysql > use nome-db;

**mysql > CREATE TABLE nometabella (id smallint unsigned not null
auto_increment, name varchar(20) not null, constraint pk_example primary key
(id));**

mysql > INSERT INTO nometabella (id, name) VALUES (null, 'Sample data');

mysql > exit



INSTALLARE PHP 7.2 E VERIFICARE LA PRESENZA DELL'INTERPRETE

```
sudo apt install php  
php -v
```



ESERCIZIO 9

**CREIAMO UN FILE PHP NELLA DOCUMENT ROOT
DEL SERVER VIRTUALIZZATO**



ESERCIZIO 9 – SVOLGIMENTO

sudo nano /var/www/html/index.php

AGGIUNGERE LE SEGUENTI RIGHE ALL'INTERNO:

```
<?php  
phpinfo();
```

E SALVARE (CTRL + X)



ESERCIZIO 10

**VERIFICHIAMO IL FUNZIONAMENTO DEL WEB SERVER
RICHIAMANDO IL FILE PHP CREATO**

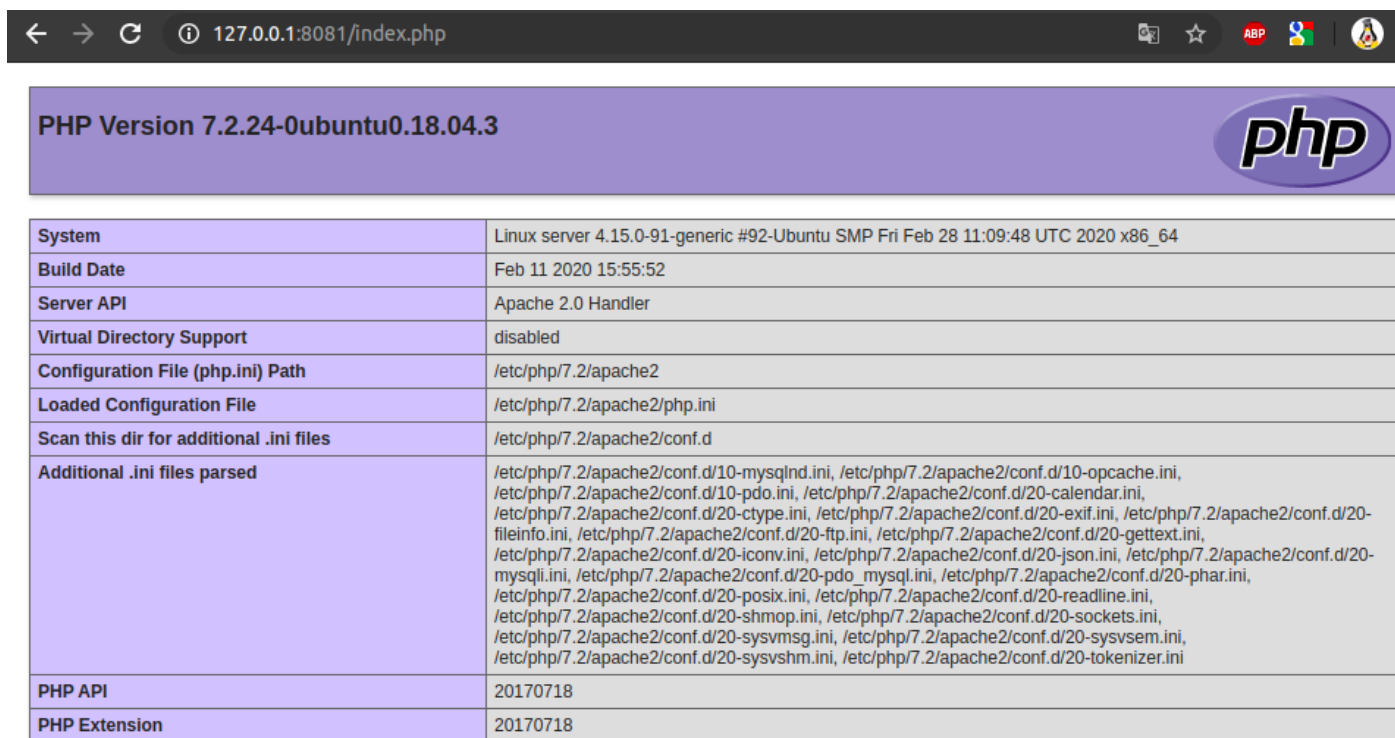
- A) DALLA MACCHINA VIRTUALE CLIENT**
- B) DAL SISTEMA HOST**



ESERCIZIO 10 (A) - SVOLGIMENTO

DA BROWSER SU VM CLIENT:

<http://10.0.2.4/index.php>



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8081/index.php`. The page content is a purple header with the text "PHP Version 7.2.24-0ubuntu0.18.04.3" and the PHP logo. Below the header is a table with system and configuration details.

System	Linux server 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64
Build Date	Feb 11 2020 15:55:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718



ESERCIZIO 10 (B) - SVOLGIMENTO

PER INIZIARE, IMPOSTIAMO IL FORWARD DELLE PORTE

PER RAGGIUNGERE LA PORTA 80 DELLA VM

SU "IP DEL GUEST" VA INSERITO L'IP DEL SERVER

Regole di inoltro delle porte

Nome	Protocollo	IP dell'host	Porta dell'host	IP del guest	Porta del guest
Rule 1	TCP	127.0.0.1	2222	10.0.2.15	22
Rule 2	TCP	127.0.0.1	8081	10.0.2.15	80


✕ Annulla ✓ OK



ESERCIZIO 10 - SVOLGIMENTO

2) DA BROWSER SU SISTEMA HOST:

`http://127.0.0.1:8081/index.php`

PHP Version 7.2.24-0ubuntu0.18.04.3		
System	Linux server 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64	
Build Date	Feb 11 2020 15:55:52	
Server API	Apache 2.0 Handler	
Virtual Directory Support	disabled	
Configuration File (php.ini) Path	/etc/php/7.2/apache2	
Loaded Configuration File	/etc/php/7.2/apache2/php.ini	
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d	
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini	
PHP API	20170718	
PHP Extension	20170718	



Domande?





PROSSIMA LEZIONE

MODULO MANUTENZIONE E SICUREZZA

**MONITORAGGIO, MANUTENZIONE ED
AGGIORNAMENTO DEL SERVER
MESSA IN SICUREZZA DEL SISTEMA**



CONTATTI

marco@orvietolinux.it

www.orvietolinux.it

f i t @orvietolinux