

First session

سوال ۱

`ls -ls`

سوال ۲

- سیستم عمومی و ضروری دستورات : `/bin`
 - سیستم مدیر مخصوص مدیریتی دستورات : `/sbin` (root)
-

سوال ۳



Second session

سوال ۱

```
mkdir -p project/assets/{css,py}  
project/pages
```

سوال ۲

```
touch txt.info  
ln txt.info txt.h_info  
ln -s txt.info txt.s_info  
rm txt.info
```

- `txt.h_info` ← میماند باقی فایل
 - `txt.s_info` ← (No such file) میشود شکسته لینک
-

سوال ۳

- حذف از قبل تأیید:

rm -i

- (تأیید بدون) اجباری حذف:

rm -f

Fifth session

۱ سوال

PID = 1

۲ سوال

SIGTSTP = 20

۳ سوال

- *SIGTERM* : پایان برای محترمانه درخواست
 - *SIGKILL* : (۹ سیگنال) مدیریت غیرقابل و اجباری پایان
-

۴ سوال

grep "word" file.txt

سوال ۵

```
dmesg | grep -Ei "error|fail" >  
log.errors_kernel
```

سوال ۶

• **tr** : تبدیل یا حذف کاراکترها

```
cat file.txt | tr 'A-Z' 'a-z' | sort
```

Sixth session

سوال ۱

در پیشفرض صورت به (.) فعلی دایرکتوری چون
نیست PATH

سوال ۲

```
#!/bin/bash  
read -p "Enter length: " l  
read -p "Enter width: " w  
echo "Perimeter = $((2*(l+w)))"  
echo "Area = $((l*w))"
```

سوال ۳

رشته یک صورت به آرگومانها همه → "\$*"
جداگانه آرگومانهای صورت به آرگومانها همه → "\$@"

Seventh session

سوال ۱

```
#!/bin/bash

if [[ $# -ne 1 ]]; then
    echo "Error"
    exit 1
fi

if [[ -e "$1" ]]; then
    echo "File exists"
else
    echo "File does not exist"
fi
```

سوال ۲

یا خالی متغیر وقتی شرط خطای از جلوگیری برای
است فاصله دارای

سوال ۳

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

```
for f in *; do
```

```
    if [[ -f "$f" ]]; then
```

```
        echo "$f : file"
```

```
    elif [[ -d "$f" ]]; then
```

```
        echo "$f : directory"
```

```
    fi
```

```
done
```


Eighth session

سوال ۱

object (.o) فایل و میکند کامپایل فقط را فایل
لینک میسازد، بدون

سوال ۲

next → ورود بدون تابع از عبور
step → تابع داخل به ورود

سوال ۳

main.c

```
#include <stdio.h>
```

```
void helper();
```

```
int main() {
```

```
    helper();
```

```
    return 0;  
}
```

helper.c

```
#include <stdio.h>  
  
void helper() {  
    printf("Hello\n");  
}
```

Makefile

```
CC=gcc
```

```
all: app
```

```
app: main.o helper.o
```

```
$(CC) main.o helper.o -o app
```

```
main.o: main.c
```

`$(CC) -c main.c`

`helper.o: helper.c`

`$(CC) -c helper.c`

Ninth session

سوال ۱

fork در والد *PID* فرزند و در فرزند مقدار ۰ برمیگرداند.

سوال ۲

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    char cmd[256];
    while (1) printf("myshell> ");
    if (!fgets(cmd, sizeof(cmd), stdin)) break;
    cmd[strcspn(cmd, "\n")] = 0;

    pid_t pid = fork();
    if (pid == 0) {
        char *args[] = {cmd, NULL};
        execvp(args[0], args);
        perror("execvp");
        exit(1);
    }
    else {
        wait(NULL);
    }
}

return 0;
}
```

سوال ۳

یا *init* توسط و میشود (*Orphan*) یتیم فرآیند
میشود سرپرستی (*PID 1*) *systemd*.