Solutions: Practical Examination

Course: Concepts of Operating Systems & Software Development Methodologies

Section A: Operating Systems

Write Linux commands to create a directory, set file permissions, and copy files.

```
mkdir mydir
chmod 755 mydir
cp /path/to/source.txt mydir/
cp ./source.txt ../backup/
```

Shell script to check whether a number is prime.

```
#!/bin/bash
read -p "Enter a number: " num
flag=0
for ((i=2; i<=num/2; i++))
do
  if [ $((num%i)) -eq 0 ]
  then
    flag=1
    break
  fi
done
if [ $flag -eq 0 ]; then
  echo "$num is prime"
else
  echo "$num is not prime"
fi
```

Shell script to display the largest of three numbers.

```
#!/bin/bash
echo "Enter three numbers:"
read a b c
if [ $a -ge $b ] && [ $a -ge $c ]; then
  echo "$a is largest"
elif [ $b -ge $a ] && [ $b -ge $c ]; then
  echo "$b is largest"
else
  echo "$c is largest"
fi
```

Shell script to print Fibonacci series.

```
#!/bin/bash
read -p "Enter n: " n
a=0
b=1
echo "Fibonacci Series:"
for (( i=0; i<n; i++ ))
do
  echo -n "$a "
  fn=$((a + b))
  a=$b
  b=$fn
```

```
done
```

Simulate FCFS scheduling algorithm.

Processes: P1=0,5; P2=1,3; P3=2,8

Turnaround Time = Completion Time - Arrival Time

Waiting Time = Turnaround Time - Burst Time

FCFS Order: P1 -> P2 -> P3

Avg Turnaround Time = (5+7+13)/3

Avg Waiting Time = (0+4+5)/3

Program using fork().

```
#include <stdio.h>
#include <unistd.h>
int main() {
  pid_t pid = fork();
  if (pid == 0) {
    printf("Child process. PID: %d\n", getpid());
  } else {
    printf("Parent process. PID: %d\n", getpid());
  }
  return 0;
}
```

Case statement calculator.

```
#!/bin/bash
echo "Enter two numbers:"
read a b
echo "Choose operation: 1.Add 2.Subtract 3.Multiply 4.Divide"
read choice
case $choice in
  1) echo "Sum = $((a+b))";;
  2) echo "Diff = $((a-b))";;
  3) echo "Prod = $((a*b))";;
  4) echo "Div = $((a/b))";;
  *) echo "Invalid";;
esac
```

Page replacement FIFO example.

Reference string: 7, 0, 1, 2, 0, 3, 0, 4

Frames = 3

FIFO Replacements = 5

Section B: Software Development Methodologies

Git repository operations.

```
git init
git add .
git commit -m "Initial commit"
git branch feature
git checkout feature
# make changes
git commit -am "Feature work"
git checkout main
git merge feature
```

Git commit history and revert.

```
git log --oneline
git revert <commit_id>
git push origin main
```

Dockerfile and run container.

```
# Dockerfile
FROM python:3.9
COPY . /app
WORKDIR /app
CMD ["python", "app.py"]

# Build and Run
docker build -t myapp .
docker run -d -p 5000:5000 myapp
```

Deploy app on Kubernetes.

```
kubectl create deployment webapp --image=nginx
kubectl expose deployment webapp --type=NodePort --port=80
kubectl get svc
```

Selenium login script.

```
from selenium import webdriver
driver = webdriver.Chrome()
driver.get("https://example.com/login")
driver.find_element("id","username").send_keys("user")
driver.find_element("id","password").send_keys("pass")
driver.find_element("id","login").click()
```

Selenium checkboxes.

```
checkboxes = driver.find_elements("css selector", "input[type='checkbox']")
for cb in checkboxes:
    cb.click()
    assert cb.is_selected()
```

Jenkins pipeline job (Jenkinsfile).

```
pipeline {
  agent any
  stages {
    stage('Build') { steps { sh 'mvn clean install' } }
    stage('Test') { steps { sh 'mvn test' } }
  }
}
```

Sample Test Plan.

Test Plan: Web Application

Test Cases:

1. Verify login with valid credentials.
2. Verify login with invalid credentials.
3. Verify navigation menu links.
4. Verify form submission with valid data.
5. Verify error messages for invalid input.