1.fcfs

```
#!/bin/bash

echo "FCFS"
read -p "Enter number of processes: " n

for ((i=0;i<n;i++)); do
  read -p "BT P$((i+1)): " bt[i]
done

wt[0]=0
for ((i=1;i<n;i++)); do
  wt[i]=$((wt[i-1]+bt[i-1]))
done

echo -e "\nPID\tBT\tWT\tTAT"
for ((i=0;i<n;i++)); do
  tat[i]=$((wt[i]+bt[i]))
  echo -e "$((i+1))\t${bt[i]}\t${wt[i]}\t${tat[i]}"
done
```

2. **Implementation of Deadlocks**

```
# -------- Process 1 --------
#!/bin/bash

echo "Process 1: Trying to lock Resource 1"
```

```
exec 10>R1.lock
flock 10
echo "Process 1: Locked Resource 1"

sleep 3

echo "Process 1: Trying to lock Resource 2"
exec 11>R2.lock
flock 11
echo "Process 1: Locked Resource 2"

echo "Process 1: Completed"
```

3.sequential file access

```
#!/bin/bash

echo "SEQUENTIAL FILE ACCESS"
read -p "Enter filename: " file

if [ ! -f "$file" ]; then
  echo "File not found!"
  exit
fi

count=1
while read line
do
  echo "Line $count: $line"
```

```
  count=$((count+1))
done < "$file"
```

4.clock synchronization

```
#!/bin/bash
echo "CLOCK SYNCHRONIZATION (Berkeley Algorithm)"

read -p "Enter number of machines: " n

sum=0
for ((i=1;i<=n;i++)); do
  read -p "Time of Machine $i (seconds): " t[i]
  sum=$((sum + t[i]))
done

avg=$((sum / n))
echo "Average Time: $avg seconds"
echo "Time adjustments for each machine:"

for ((i=1;i<=n;i++)); do
  adj=$((avg - t[i]))
  echo "Machine $i: $adj seconds"
done

echo "Synchronization Completed!"
```

5.posix real time threads

```
#!/bin/bash
echo "=== POSIX Real-Time Threads Simulation ==="
```

```
# High Priority Task (FIFO)
high_task() {
  for i in {1..5}; do
    echo "High Priority Task running..."
    sleep 1
  done
}

# Low Priority Task (Round Robin)
low_task() {
  for i in {1..5}; do
    echo "Low Priority Task running..."
    sleep 1
  done
}

# Run tasks with simulated priorities
nice -n -5 bash -c 'for i in {1..5}; do echo "High Priority Task running..."; sleep 1; done' &
nice -n 10 bash -c 'for i in {1..5}; do echo "Low Priority Task running..."; sleep 1; done' &

wait
echo "All simulated real-time tasks completed!"
```

6.real time simulation

```
#!/bin/bash

echo "=== Real-Time Simulation ==="
```

```
task1() {
  for i in {1..5}; do
    echo "Task 1 (High Priority) executing at $(date +%T)"
    sleep 1
  done
}

task2() {
  for i in {1..5}; do
    echo "Task 2 (Low Priority) executing at $(date +%T)"
    sleep 1
  done
}
task1 &
task2 &

wait
echo "All tasks completed! Real-Time Simulation Done."
```