

Operating System CA-2

Name: Arya Manoj Madhavi

Div: D10B

Roll no: 30

1. To write a C program for implementation of FIFO page replacement algorithm.

CODE:

```
#include < stdio.h >
int main()
{
    int incomingStream[] = {4 , 2 , 1 , 4 , 3};
    int pageFaults = 0;
    int frames = 3;
    int m, n, s, pages;
    pages = sizeof(incomingStream)/sizeof(incomingStream[0]);
    printf(" Incoming \ t Frame 1 \ t Frame 2 \ t Frame 3 ");
    int temp[ frames ];
    for(m = 0; m < frames; m++)
    {
        temp[m] = -1;
    }
    for(m = 0; m < pages; m++)
    {
        s = 0;
        for(n = 0; n < frames; n++)
        {
            if(incomingStream[m] == temp[n])
            {
                s++;
                pageFaults--;
            }
        }
        pageFaults++;
    }
```

```

if((pageFaults <= frames) && (s == 0))
{
    temp[m] = incomingStream[m];
}
else if(s == 0)
{
    temp[(pageFaults - 1) % frames] = incomingStream[m];
}
printf("\n");
printf("%d\t\t\t",incomingStream[m]);
for(n = 0; n < frames; n++)
{
    if(temp[n] != -1)
        printf(" %d\t\t\t", temp[n]);
    else
        printf(" - \t\t\t");
}
}
printf("\nTotal Page Faults:\t%d\n", pageFaults);
return 0;
}

```

OUTPUT:

```

D:\fiffo.exe
Incoming  t Frame 1  t Frame 2  t Frame 3
4         4         -         -
2         4         2         -
1         4         2         1
4         4         2         1
3         3         2         1
Total Page Faults:      4

Process returned 0 (0x0)   execution time : 0.166 s
Press any key to continue.
|

```

2. Implement various disk scheduling algorithms like SCAN, and C-SCAN in C/Python/Java.

CODE:

```
# Python3 program to demonstrate C-SCAN Disk Scheduling algorithm
size = 8
disk_size = 200
```

```
def CSCAN(arr, head):
    seek_count = 0
    distance = 0
    cur_track = 0
    left = []
    right = []
    seek_sequence = []

    # Appending end values
    left.append(0)
    right.append(disk_size - 1)

    # Tracks on the left of the
    for i in range(size):
        if (arr[i] < head):
            left.append(arr[i])
        if (arr[i] > head):
            right.append(arr[i])

    # Sorting left and right vectors
    left.sort()
    right.sort()

    # First service the requests
    for i in range(len(right)):
        cur_track = right[i]

        # Appending current track
        seek_sequence.append(cur_track)

        # Calculate absolute distance
        distance = abs(cur_track - head)

        # Increase the total count
        seek_count += distance

        # Accessed track is now new head
        head = cur_track

    # Once reached the right end
    # jump to the beginning.
```

```

head = 0

# adding seek count for head returning from 199 to 0
seek_count += (disk_size - 1)

# Now service the requests again
# which are left.
for i in range(len(left)):
    cur_track = left[i]

    # Appending current track
    # to seek sequence
    seek_sequence.append(cur_track)

    # Calculate absolute distance
    distance = abs(cur_track - head)

    # Increase the total count
    seek_count += distance

    # Accessed track is now the new head
    head = cur_track

print("Total number of seek operations =",
      seek_count)
print("Seek Sequence is")
print(*seek_sequence, sep="\n")

# Driver code
# request array
arr = [30, 37, 42, 43,
       44, 46, 48, 51]
head = 40
print("Initial position of head:", head)
CSCAN(arr, head)

```

OUTPUT:

```

PS C:\Users\aryam> & C:/Users/aryam/AppData/Local/Programs/Python/Python312/python.exe d:/scan.py
Initial position of head: 40
Total number of seek operations = 395
Seek Sequence is
42
43
44
46
48
51
199
0
30
37
PS C:\Users\aryam>

```

3. Case Study on Cloud and IoT Operating Systems

Introduction:

Cloud computing and the Internet of Things (IoT) have revolutionized the way businesses operate, offering unprecedented scalability, flexibility, and connectivity. This case study delves into the operating systems (OS) used in cloud and IoT environments, examining their functionalities, challenges, and impacts on businesses. The integration of Internet of Things (IoT) technologies has also revolutionized various aspects of daily life, including home automation. This case study delves into the application of IoT operating systems (OS) in enabling smart home automation, showcasing how these systems enhance convenience, efficiency, and security for homeowners.

1. Cloud Operating Systems:

In the realm of cloud computing, operating systems play a pivotal role in managing virtualized resources and facilitating seamless interactions between users and cloud infrastructure. Popular cloud OS options include Linux-based systems like Ubuntu Server and CentOS, as well as specialized distributions such as OpenStack for managing private clouds. These OS prioritize scalability, security, and resource optimization to support diverse workloads in the cloud environment.

2. IoT Operating Systems:

IoT operating systems cater to the unique requirements of connected devices, emphasizing low resource consumption, real-time data processing, and interoperability. Examples of IoT OS include FreeRTOS, RIOT OS, and Zephyr, which are designed to run on constrained devices with minimal memory and processing capabilities. These OS enable efficient communication between IoT devices, cloud platforms, and end-users, facilitating the seamless exchange of data and insights.

Features :

1. IoT Operating System:

a. **Resource Efficiency:** IoT operating systems are designed to run on resource-constrained devices with limited processing power and memory, optimizing resource utilization for efficient operation.

b. Real-time Processing: These OS prioritize real-time data processing capabilities, enabling quick analysis and response to incoming data streams from IoT sensors and devices.

c. Connectivity: IoT OS provides robust support for various communication protocols such as MQTT, CoAP, and Bluetooth Low Energy (BLE), facilitating seamless connectivity and interoperability between devices.

d. Security: Security features are integrated into IoT operating systems to protect sensitive data, authenticate devices, and mitigate cybersecurity threats in interconnected IoT environments.

e. Scalability: IoT OS are scalable to accommodate growing numbers of connected devices, ensuring smooth operation and management of large-scale IoT deployments.

2. Cloud Operating System:

a. Virtualization: Cloud operating systems leverage virtualization technologies to abstract and pool resources from underlying hardware, enabling on-demand provisioning and allocation of computing, storage, and networking resources.

b. Scalability: These OS are designed for horizontal scalability, allowing cloud infrastructure to dynamically scale up or down in response to changing workload demands, ensuring high availability and performance.

c. Multi-tenancy: Cloud OS supports multi-tenancy, enabling multiple users or organizations to securely share the same underlying infrastructure while maintaining isolation and data privacy.

d. Automation: Cloud operating systems incorporate automation and orchestration tools for deploying, managing, and monitoring cloud resources, streamlining IT operations and reducing manual intervention.

e. Security: Robust security measures such as encryption, identity and access management (IAM), and network segmentation are integrated into cloud OS to protect data and applications from unauthorized access and cyber threats.

Integration Challenges:

While cloud and IoT operating systems excel in their respective domains, integrating them presents challenges due to differences in architecture, communication protocols, and security requirements. Bridging the gap between cloud and IoT OS necessitates robust middleware solutions, standardized protocols (e.g., MQTT, CoAP), and secure communication channels to ensure interoperability and data integrity across the ecosystem.

Real-Life Applications :

1. IoT Operating Systems:

1. Smart Home Automation: IoT operating systems power smart home devices such as thermostats, lighting systems, and security cameras, enabling users to remotely monitor and control their home environment.

Smart home automation powered by IoT operating systems finds diverse applications in enhancing convenience, comfort, and security for homeowners:

- a. Automated Lighting: IoT-enabled light bulbs and switches controlled by smart home hubs or voice assistants allow homeowners to remotely adjust lighting levels, schedule lighting routines, and conserve energy.
- b. Intelligent Thermostats: Smart thermostats equipped with IoT technology adapt heating and cooling settings based on occupancy patterns, weather forecasts, and user preferences, optimizing energy usage and maximizing comfort.
- c. Home Security Systems: IoT-operated security cameras, doorbell cameras, and motion sensors provide real-time surveillance and alerts, enabling homeowners to monitor their property remotely and deter potential intruders.
- d. Smart Appliances: IoT-enabled appliances such as refrigerators, ovens, and washing machines offer advanced features like remote monitoring, energy usage tracking, and automated maintenance notifications, enhancing efficiency and convenience.
- e. Environmental Monitoring: IoT sensors deployed throughout the home monitor air quality, humidity levels, and carbon dioxide concentrations, providing insights to optimize indoor environmental conditions for health and comfort.

In conclusion, smart home automation powered by IoT operating systems represents a paradigm shift in residential living, offering homeowners unprecedented control, convenience, and peace of mind. By leveraging IoT technologies, homeowners can transform their living spaces into intelligent ecosystems that adapt to their preferences, enhance efficiency, and promote sustainability. As the adoption of smart home devices

continues to rise, investing in robust IoT operating systems and interoperable solutions will be crucial for realizing the full potential of smart home automation in the digital age.

2. Industrial IoT (IIoT): In industrial settings, IoT OS facilitates the monitoring and optimization of manufacturing processes, predictive maintenance of equipment, and real-time tracking of inventory and assets.

3. Healthcare: IoT operating systems support the development of remote patient monitoring systems, wearable health trackers, and medical devices for managing chronic conditions and improving healthcare outcomes.

5. Smart Agriculture: In agriculture, IoT OS are utilized in precision farming solutions for monitoring soil conditions, irrigation systems, and crop health, leading to increased productivity and resource efficiency.

2. Cloud Operating Systems:

a. Software as a Service (SaaS): Cloud operating systems power SaaS applications such as customer relationship management (CRM), enterprise resource planning (ERP), and collaboration tools, delivering on-demand software solutions to users over the internet.

b. Infrastructure as a Service (IaaS): In the IaaS model, cloud OS enables organizations to provision and manage virtualized infrastructure components such as virtual machines (VMs), storage, and networking resources on a pay-as-you-go basis.

c. Platform as a Service (PaaS): Cloud operating systems support PaaS offerings that provide developers with tools and frameworks for building, deploying, and scaling applications without worrying about underlying infrastructure management.

d. Big Data Analytics: Cloud OS are instrumental in supporting big data analytics platforms and services, providing scalable computing and storage resources for processing and analyzing large volumes of data to derive valuable insights.

e. Disaster Recovery and Business Continuity: Cloud operating systems enable organizations to implement disaster recovery and business continuity strategies by replicating data and applications across geographically distributed cloud regions, ensuring resilience and data redundancy.

Conclusion:

In conclusion, the development in cloud and IoT operating systems offers immense potential for innovation and growth, empowering businesses to harness the full potential of connected technologies. As organizations navigate the complexities of hybrid environments, investing in robust OS, interoperability standards, and strategic partnerships will be critical to realizing the promise of cloud and IoT integration in the digital era.

This case study highlights the significance of operating systems in shaping the future of cloud and IoT ecosystems, driving technological advancements, and fueling business transformation. IoT and cloud operating systems offer a wide range of features and applications that drive innovation, efficiency, and agility across various industries, paving the way for the widespread adoption of connected technologies and cloud computing paradigms in the digital age.