CAPSTONEPROJECTREPORT

ProjectTitle: Bash Scripting Suite for System Maintenance

Project Code: Project 5 - LinuxOS and Shell Programming

Submitted by:

StudentName:AdarshPujari

RollNo: 2241016496

DepartmentofComputerScience&Engineering Institute

Of Technical Education and Research

1. Abstract

ThisprojectimplementsanautomatedSystemMaintenanceSuiteusingBashscriptingfor Linux environments.
Thesuiteautomatessystem-leveltaskssuchasperformingdatabackups,runningupdates and cleanup, monitoring logs,
andmanagingalloperationsthroughauser-friendlymenu-driveninterface.Thesescripts collectively aim to simplify
andautomatesystem administration,improvingefficiency,security,andmaintainability.

2. Objectives

- ToautomatedailyandperiodicLinuxsystemmaintenance tasks.

- Toimplementmodularscripts forbackup,cleanup,update,andlog monitoring.

- Tocreateacentralinteractiveinterfaceforexecutingalloperations.

- ToensurereliabilityandreusabilityacrossLinuxandWSLenvironments.

- Tomaintaincomprehensiveloggingandsecurityforeveryoperation.

3. SystemOverview

TheSystemMaintenanceSuiteiscomposedofmultipleBashscriptsthatautomateand streamline Linux administration.
Eachscriptperformsaspecificmaintenanceoperation,andtogethertheyformacomplete toolkit for maintaining system
health.Thescriptsareintegratedunderacentralmenu(`maintenance_menu.sh`),allowing system administrators to perform
taskseasilysuchasrunningbackups,checkinglogs,cleaningthesystem,orupdating software packages. The suite also
includesasetupscript(`install.sh`)thatconfigurestheenvironmentandoptionallysetsupa daily cron job for automation.

4. Implementation(Code)

backup.sh

```
#!/usr/bin/envbash
#============================================================
#backup.sh-AutomatedBackupScript(Day1-Assignment5)
```

```
#============================================================
#Usage:
#     sudo./backup.sh/path/to/source[another/source...] #
#Description:
#Createsa timestamped compressedarchive(.tar.gz)
#ofthegivendirectories/filesandsavesittothe #backup
directory.Keeps only the latest7 backups.#
#============================================================

set-euopipefail
IFS=$'\n\t'

#---Configuration---
BACKUP_DIR="/var/backups/system-maintenance-
suite"LOG_DIR="$(dirname"$0")/../logs"LOGFILE="$
LOG_DIR/backup.log"
RETENTION_COUNT=7   #Keeplast7backups

#---Functions---
timestamp(){date'+%Y-%m-%d_%H-%M-%S';}

log(){
     mkdir-p"$LOG_DIR"
     echo"$(timestamp)   $*"|tee-a"$LOGFILE"
}

error_exit(){
     log"ERROR:$1"e
     xit1
}

#---Validations---
if["$#"-lt1];then
     echo"Usage:$0/path/to/source[another/source...]"exit
     2
fi

if["$EUID"-ne0];then
     echo"Pleaserunasroot(sudo)toaccessalldirectories."exit3
fi

#---PrepareBackupDirectory---
sudomkdir-p"$BACKUP_DIR"
sudochmod700"$BACKUP_DIR"

SRC_LIST=("$@")
SAFE_NAME=$(printf"%s_""${SRC_LIST[@]##*/}"|sed's/[^A-Za-z0-9_-]//g'|sed
's/_$//')
ARCHIVE_NAME="${SAFE_NAME}_$(timestamp).tar.gz"ARCHIVE_PATH="$BA
CKUP_DIR/$ARCHIVE_NAME"

log"Startingbackupof:${SRC_LIST[*]}"
```

```
#---Validateeachsourcepath---
forpathin"${SRC_LIST[@]}";do
    if[!-e"$path"];then
        error_exit"Sourcepathnotfound:$path"
    fi
done

#---CreateBackup(correcttarorder)---
iftar--warning=no-file-changed--ignore-failed-read\
        --exclude=/proc--exclude=/sys--exclude=/dev\
        -czf"$ARCHIVE_PATH""${SRC_LIST[@]}"2>>"$LOGFILE";then
    chmod600"$ARCHIVE_PATH"
    SIZE=$(du-h"$ARCHIVE_PATH"|cut-f1)
    log"Backupcreated:$ARCHIVE_PATH($SIZE)"else
    error_exit"Tarcommandfailedwhilebackingup${SRC_LIST[*]}"
fi

#---Rotateoldbackups---
mapfile-tfiles<<(ls-1t"$BACKUP_DIR"/"${SAFE_NAME}"_*.tar.gz2>/dev/null||true)
if["${#files[@]}"-gt"$RETENTION_COUNT"];then
    to_delete=("${files[@]:$RETENTION_COUNT}")
    forfin"${to_delete[@]}";do
        rm-f--"$f"&&log"Removedoldbackup:$f"done
fi

log"Backupcompletedsuccessfullyfor:${SRC_LIST[*]}"echo"
✓Backupcomplete.Archivesavedto:$ARCHIVE_PATH"exit0
```

system_update_and_cleanup.sh

```
#!/usr/bin/envbash
#============================================================
#system_update_and_cleanup.sh-Day2:SystemMaintenance
#============================================================
#Usage:
#    sudo./system_update_and_cleanup.sh[--dry-run] #
#Description:
#    Updatesthesystempackages,removesunnecessaryfiles,
#    cleanscaches,rotatesoldlogs,andrecordsallactions. #
    Use--dry-runtosimulateactionssafely.
#
#============================================================

set-euopipefail
IFS=$'\n\t'

#---Configuration---
```

```
LOG_DIR="$(dirname"$0")/../logs"LOGFILE="$LOG_DIR/system_u
pdate.log"
DRY_RUN=false

#---Functions---
timestamp(){date'+%Y-%m-%d_%H-%M-%S';}

log(){
    mkdir-p"$LOG_DIR"
    echo"$(timestamp)  $*"|tee-a"$LOGFILE"
}

run_cmd(){
    localcmd="$1"
    if["$DRY_RUN"=true];then
        log"[DRY-RUN]Wouldexecute:$cmd"else
        log"Running:$cmd"
        eval"$cmd">>"$LOGFILE"2>&1||log"Warning:commandfailed-$cmd"
    fi
}

#---Parsearguments---
if["${1:-}"="--dry-run" ];then
    DRY_RUN=true
fi

#---Safetycheck---
if["$EUID"-ne0];then
    echo"Pleaserunasroot(sudo)."
    exit1
fi

log"===Startingsystemupdateandcleanup(dry-run=$DRY_RUN)==="#---

1.Updatepackagelistsandupgrade---
run_cmd"aptupdate-y"
run_cmd"aptupgrade-
y"run_cmd"aptfull-upgrade-y"

#---2.Removeunnecessarypackagesandcleancaches---
run_cmd"aptautoremove-y"
run_cmd"aptautoclean-
y"run_cmd"aptclean-y"

#---3.Rotateorcompressoldaptlogs---
APT_LOG_DIR="/var/log/apt"
if[-d"$APT_LOG_DIR"];then
    run_cmd"find$APT_LOG_DIR-typef-name'*.log.*'-mtime+14-
    delete"run_cmd"gzip-f$APT_LOG_DIR/*.log||true"
    log"Aptlogscleanedandcompressed(olderthan14daysremoved)."else
    log"Aptlogdirectorynotfoundat$APT_LOG_DIR."
fi
```

```
# --- 4.Cleargeneralsystemlogsolderthan30days(optional) ---
run_cmd"find/var/log -type f -name'*.log' -mtime+30 -execrm -f{}+"

#---5.Updatesysteminformationdatabase---
run_cmd"updatedb"

log"===Systemupdateandcleanupcompletedsuccessfully==="
echo"✓Systemupdateandcleanupcomplete.Check$LOGFILEfordetails."exit0
```

log_monitor.sh

```
#!/usr/bin/envbash
#============================================================
#log_monitor.sh-Day3:LogMonitoringandAlerting
#============================================================
#Usage:
#    sudo./log_monitor.sh#
#Description:
#    Scanskeysystemlogsforerrors,warnings,andfailedlogins. #
     Generatesasummaryreportinthelogsdirectory.
#
#============================================================

set-euopipefail
IFS=$'\n\t'

#---Configuration---
LOG_DIR="$(dirname"$0")/../logs"
REPORT_FILE="$LOG_DIR/log_monitor_report.txt"MAIN_LOG="$LOG_DI
R/log_monitor.log"

#Logstoscan(commonDebian/Ubuntulocations) LOG_FILES=(
  "/var/log/syslog""/var/log/auth.lo
  g""/var/log/kern.log"
)

#Keywordstodetect KEYWORDS=(
  "error""failed
  ""critical""una
  uthorized""den
  ied""panic""s
  egfault"
)

#---Functions---
timestamp(){date'+%Y-%m-%d_%H-%M-%S';}
```

```
log(){
    mkdir-p"$LOG_DIR"
    echo"$(timestamp)  $*"|tee-a"$MAIN_LOG"
}

#---Safetycheck---
if["$EUID"-ne0];then
    echo"Pleaserunasroot(sudo)."
    exit1
fi

log"===Startinglogmonitoring==="#--

-Initializereport---
echo"====SystemLogMonitoringReport($(timestamp))====">"$REPORT_FILE"
echo>>"$REPORT_FILE"

#---Scaneachlogfile---
forfilein"${LOG_FILES[@]}";do
    if[-f"$file" ];then
        echo"Analyzing:$file">>"$REPORT_FILE"fork
        eywordin"${KEYWORDS[@]}";do
            matches=$(grep-i"$keyword""$file"|tail-n10||true) if[-
            n"$matches" ];then
                echo"----Matchesfor'$keyword'----
                ">>"$REPORT_FILE"echo"$matches">>"$REPORT_FILE"
                echo>>"$REPORT_FILE"
            fi
        done
        echo" ----------">>"$REPORT_FILE"
    else
        echo"Logfilenotfound:$file">>"$REPORT_FILE"
    fi
done

#---Summary---
echo>>"$REPORT_FILE"
echo"====EndofReport====">>"$REPORT_FILE"

log"Monitoringcompleted.Reportsavedat$REPORT_FILE"
echo"✅Logmonitoringcomplete.Check:$REPORT_FILE"exit0
```

maintenance_menu.sh(Updated)

```
#!/usr/bin/envbash
#============================================================
#maintenance_menu.sh-
UnifiedMaintenanceDashboard(Fixed)#==========================
==================================
#Usage:
#    sudo./maintenance_menu.sh#
#Description:
#    Providesaninteractivemenutorun:
```

```
#       -Backup
#       -SystemUpdate&Cleanup #
        -LogMonitoring
#       -ViewLogs&Reports
#============================================================

set-euopipefail
IFS=$'\n\t'

#---Configuration---
SCRIPT_DIR="$(cd"$(dirname"${BASH_SOURCE[0]}")"&&pwd)"LOG_D
IR="$SCRIPT_DIR/../logs"BACKUP_SCRIPT="$SCRIPT_DIR/backup.sh"U
PDATE_SCRIPT="$SCRIPT_DIR/system_update_and_cleanup.sh"MONITOR_S
CRIPT="$SCRIPT_DIR/log_monitor.sh"

#---UtilityFunctions---
timestamp(){date'+%Y-%m-%d_%H-%M-%S';}

pause(){
    echo
    read-rp"PressEntertocontinue..."
}

check_root(){
    if["$EUID"-ne0];then
        echo"Pleaserunasroot(sudo)." exit1
    fi
}

log(){
    mkdir-p"$LOG_DIR"
    echo"$(timestamp)  $*"|tee-a"$LOG_DIR/menu.log"
}

#---MenuFunctions---
run_backup(){
    echo
    read-rp"Enterthedirectorypath(s)tobackup(space-separated):"path_list
    if[-z"$path_list"];then
        echo"Nopathsentered.Returningtomenu."return
    fi

    log"Userinitiatedbackupfor:$path_list"

    #✓FIX:Properlysplitinputintoanarrayformultiplepaths
    IFS=''read-r-apath_array<<<"$path_list"

    #Runbackupscriptwitharray-expandedarguments
    sudo"$BACKUP_SCRIPT""${path_array[@]}"

    pause
```

```
}

run_update(){
    echo
    read-rp"Runindry-runmodefirst?(y/n):"choice if
    [["$choice"=~^[Yy]$ ]];then
        sudo"$UPDATE_SCRIPT"--dry-run
    else
        sudo"$UPDATE_SCRIPT"
    fi
    pause
}

run_monitor(){
    log"Runninglogmonitor..."sudo"
    $MONITOR_SCRIPT"
    pause
}

view_logs(){
    echo
    echo"Availablelogsin$LOG_DIR:"ls-
    1"$LOG_DIR"||true
    echo
    read-rp"Enterlog/reportfilenametoview(orpressEntertocancel):"file
    if[-n"$file"];then
        if[-f"$LOG_DIR/$file"];then echo
            echo"----Showinglast50linesof$file----------"
            tail-
        n50"$LOG_DIR/$file"else
            echo"Filenotfound:$LOG_DIR/$file"
        fi
    fi
    pause
}

#---MainMenuLoop--- check_root
mkdir-p"$LOG_DIR"

whiletrue;do
    clear
    echo"============================================="
    echo"□   SystemMaintenanceSuite    (Day4-
    Fixed)"echo"=========================================
    ======"
    echo"1□    RunBackup"
    echo"2□
                RunSystemUpdate&Cleanup"ec
    ho"3□      RunLogMonitoring"
    echo"4□
                ViewLogs/Reports"ec
    ho"5□      Exit"
    echo"============================================="
    read-rp"Chooseanoption[1-5]:"choice
```

```
    case"$choice"in
        1) run_backup;;
        2) run_update;;
        3) run_monitor;;
        4) view_logs;;
        5)
            echo"ExitingMaintenanceMenu.Goodbye!"log"
            Menuexitedbyuser."
            exit0
            ;;
        *)
            echo"Invalidoption.Tryagain." sleep1
            ;;

    esac
done
```

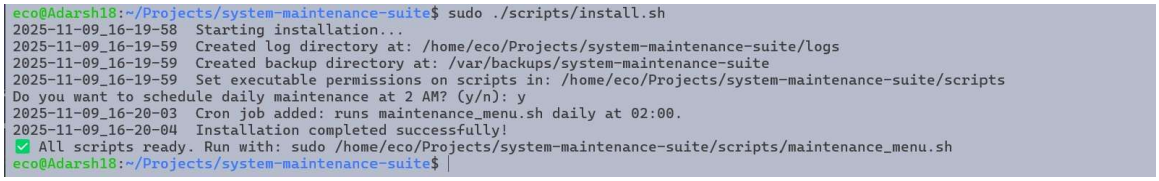
install.sh(Updated)

```
#!/usr/bin/envbash
#============================================================
#install.sh-SetupandAutomation(Fixedversion)
#============================================================
#Usage:
#    sudo./install.sh
#
#Worksbothfromtheprojectrootorfromwithinthescripts/folder.
#============================================================

set-euopipefail
IFS=$'\n\t'

#---Detectbasedirectories---
SCRIPT_PATH="$(cd"$(dirname"${BASH_SOURCE[0]}")"&&pwd)"if
[["$(basename"$SCRIPT_PATH")"=="scripts" ]];then
    BASE_DIR="$(dirname"$SCRIPT_PATH")"
    SCRIPTS_DIR="$SCRIPT_PATH"
else
    BASE_DIR="$SCRIPT_PATH"
    SCRIPTS_DIR="$BASE_DIR/scripts"
fi

LOG_DIR="$BASE_DIR/logs"BACKUP_DIR="/var/backups
/system-maintenance-suite"

timestamp(){date'+%Y-%m-%d_%H-%M-%S';}
log(){echo"$(timestamp)    $*";}

#---SafetyCheck---
if["$EUID"-ne0];then
    echo"Pleaserunasroot(sudo)."
    exit1
fi
```

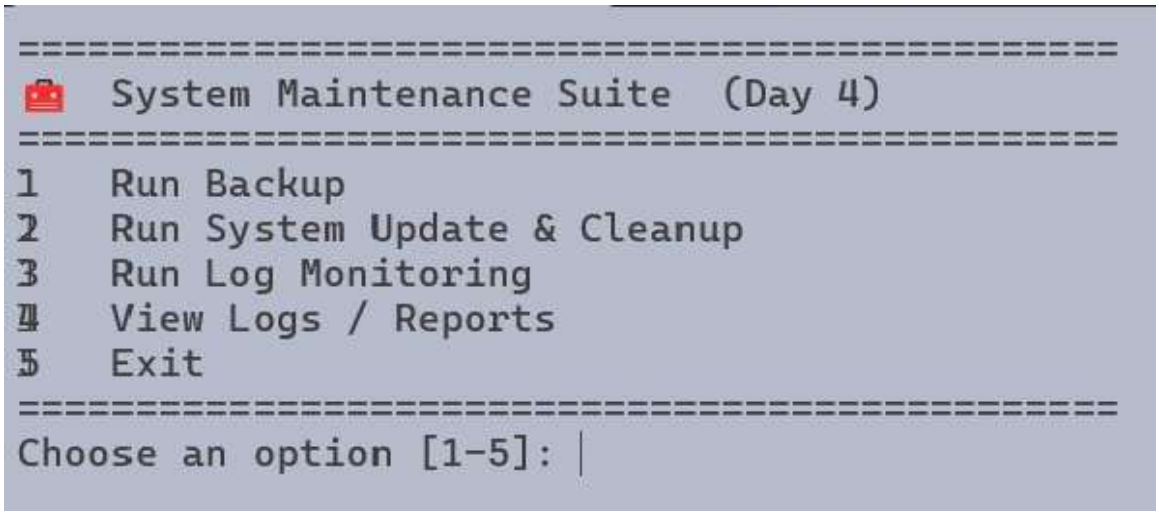
```
log"Startinginstallation..."

#---Createdirectories---mkdir-
p"$LOG_DIR""$BACKUP_DIR"
chmod700"$BACKUP_DIR"

log"Createdlogdirectoryat:$LOG_DIR"
log"Createdbackupdirectoryat:$BACKUP_DIR"

#---Makeallscriptsexecutable---
if[-d"$SCRIPTS_DIR"];then
    chmod+x"$SCRIPTS_DIR"/*.sh
    log"Setexecutablepermissionsonscriptsin:$SCRIPTS_DIR"else
    log"WARNING:Scriptdirectorynotfoundat$SCRIPTS_DIR"
fi

#---Optionalcronsetup---
read-rp"Doyouwanttoscheduledailymaintenanceat2AM?(y/n):"choice
if[["$choice"=~^[Yy]$]];then
    (crontab-l2>/dev/null;echo"02***sudo
$SCRIPTS_DIR/maintenance_menu.sh>>$LOG_DIR/cron_run.log2>&1")|crontab-
    log"Cronjobadded:runsmaintenance_menu.shdailyat02:00."
else
    log"Cronsetupskipped."
fi

log"Installationcompletedsuccessfully!"
echo"✓Allscriptsready.    Runwith:sudo$SCRIPTS_DIR/maintenance_menu.sh"
exit0
```

5. ExecutionandOutputScreenshots

ThefollowingscreenshotsdemonstratethesuccessfulexecutionoftheSystemMaintenance Suite:



Screenshot1:Installationprocessexecutedsuccessfullyviainstall.sh.



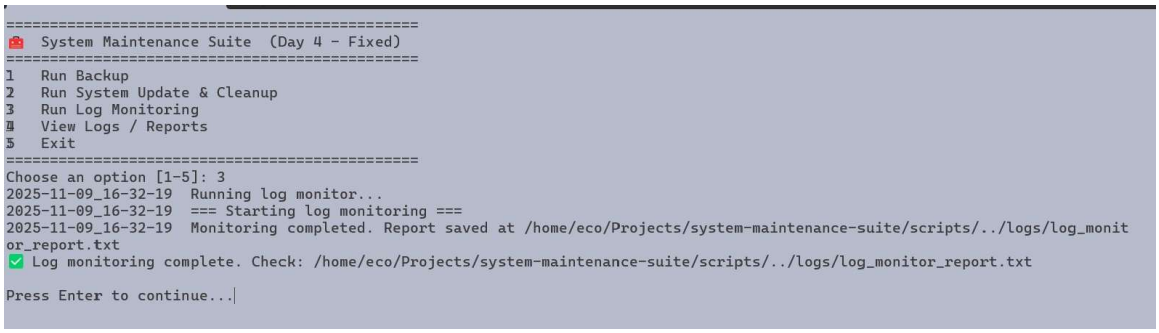
Screenshot2:DisplayoftheSystemMaintenanceSuiteinteractivemenu.

```
==============================================
🧰  System Maintenance Suite  (Day 4 - Fixed)
==============================================
1   Run Backup
2   Run System Update & Cleanup
3   Run Log Monitoring
4   View Logs / Reports
5   Exit
==============================================
Choose an option [1-5]: 1

Enter the directory path(s) to back up (space-separated): /etc /home/eco/Projects
2025-11-09_16-31-33  User initiated backup for: /etc /home/eco/Projects
2025-11-09_16-31-33  Starting backup of: /etc
/home/eco/Projects
2025-11-09_16-31-33  Backup created: /var/backups/system-maintenance-suite/etc_Projects_2025-11-09_16-31-33.tar.gz (464K)
2025-11-09_16-31-33  Backup completed successfully for: /etc
/home/eco/Projects
✅ Backup complete. Archive saved to: /var/backups/system-maintenance-suite/etc_Projects_2025-11-09_16-31-33.tar.gz

Press Enter to continue...
```

Screenshot3:Successfulexecutionofthebackup.sh script.

```
==============================================
🧰  System Maintenance Suite  (Day 4 - Fixed)
==============================================
1   Run Backup
2   Run System Update & Cleanup
3   Run Log Monitoring
4   View Logs / Reports
5   Exit
==============================================
Choose an option [1-5]: 2

Run in dry-run mode first? (y/n): n
2025-11-09_16-31-50  === Starting system update and cleanup (dry-run=false) ===
2025-11-09_16-31-50  Running: apt update -y
2025-11-09_16-31-54  Running: apt upgrade -y
2025-11-09_16-31-54  Running: apt full-upgrade -y
2025-11-09_16-31-55  Running: apt autoremove -y
2025-11-09_16-31-58  Running: apt autoclean -y
2025-11-09_16-31-59  Running: apt clean -y
2025-11-09_16-31-59  Running: find /var/log/apt -type f -name '*.log.*' -mtime +14 -delete
2025-11-09_16-31-59  Running: gzip -f /var/log/apt/*.log || true
2025-11-09_16-31-59  Apt logs cleaned and compressed (older than 14 days removed).
2025-11-09_16-31-59  Running: find /var/log -type f -name '*.log' -mtime +30 -exec rm -f {} +
2025-11-09_16-31-59  Running: updatedb
2025-11-09_16-31-59  Warning: command failed - updatedb
2025-11-09_16-31-59  === System update and cleanup completed successfully ===
✅ System update and cleanup complete. Check /home/eco/Projects/system-maintenance-suite/scripts/../logs/system_update.log for d
etails.

Press Enter to continue...
```

Screenshot4:Systemupdateandcleanupperformed successfully.



Screenshot5:Logmonitoringprocessandreport generation.

Screenshot6:SystemMaintenanceSuiteexitconfirmation message.

6. Conclusion

Project successfully automated key system maintenance tasks using Bash scripting. It included backup, system updates, and log monitoring with error handling and a user-friendly menu interface. The project enhanced understanding of Linux automation and improved scripting efficiency for real-world maintenance operations. It demonstrated effective use of shell commands for task scheduling and system monitoring. Overall, the project strengthened practical skills in Linux administration and scripting.