

```

let archivoACargar = document.getElementById("archivoACargar");
let kernel = document.getElementById("kernel");
let contenidoMemoria = document.getElementById("contenidoMemoria");
let operationSection = document.getElementById("operationSection");
let variablesSection = document.getElementById("variablesSection");
let mostrarContMemoria = document.getElementById("mostrarContenidoMemoria");
let tamMemoria = document.getElementById("memoria");
let screenContent = document.getElementById("screenContent");
let imprimir = document.getElementById("imprimir");
let overlay = document.getElementById("overlay");
let cerrarVentImprimir = document.getElementById("cerrarVentImprimir");
let printSection = document.getElementById("printSection");

//Crear la memoria con la que mostraremos Contenido y Dirección
var memoriaDirection = [];
var memoriaContenido = [];

/*Funciones*/

//Funcion para encender el Sistema Operativo haciendo la comprobación de la
memoria
function encender() {
  if ((Number(tamMemoria.value) > 9999)) {
    alert("Error, el espacio de memoria no puede exceder las 9999 posiciones
");
    apagar();
  }
  tamMemoria.disabled = true;
  kernel.disabled = true;
  archivoACargar.disabled = false;
  mostrarContMemoria.disabled = false;
  imprimir.disabled = false;
  operationSection.style.visibility = "visible";
  variablesSection.style.visibility = "visible";
  screenContent.style.background = "white";
  //Reservar la primera posición para el acumulador
  memoriaDirection.push("0000");
  memoriaContenido.push("Acumulador");
  //Verificamos que el kernel quepa
  if ((Number(tamMemoria.value)) < (Number(kernel.value) + 1)) {
    alert("Espacio de memoria excedido, insuficiente espacio para el kernel"
);
    apagar();
  }
  //Ocupar los campos de la memoria para el kernel

```

```

for (let espacio = 1; espacio < Number(kernel.value) + 1; espacio++) {
    memoriaDirection.push(zeroFill(espacio,4));
    memoriaContenido.push("***CHSOS V2021**");
}
//llenamos los array de posiciones vacias
var memoriaDirection1 = [];
var memoriaContenido1 = [];
for (var i = 0; i < memoriaDirection.length; i++) {
    memoriaDirection1.push(memoriaDirection[i]);
    memoriaContenido1.push(memoriaContenido[i]);
}
for (let campos = memoriaContenido.length; campos <= (Number(tamMemoria.value) - 1); campos++) {
    memoriaDirection1.push(zeroFill((memoriaDirection1.length),4));
    memoriaContenido1.push("-");
}
//Mostramos las posiciones de memoria con el kernel
document.getElementById("directionSection").innerHTML = memoriaDirection1.join("<br>");
document.getElementById("contentSection").innerHTML = memoriaContenido1.join("<br>");
}

function apagar() {
    tamMemoria.disabled = false;
    kernel.disabled = false;
    archivoACargar.disabled = true;
    contenidoMemoria.style.visibility = "hidden";
    operationSection.style.visibility = "hidden";
    variablesSection.style.visibility = "hidden";
    location.reload();
}

function mostrarContenidoMemoria() {
    contenidoMemoria.style.visibility = "visible";
}

archivoACargar.addEventListener("change", leerArchivo);

function leerArchivo(evento){
    for(let i=0; i<evento.target.files.length; i++) {
        let archivo = evento.target.files[i];

        //Procesamos el archivo cargado
        procesarArchivo(archivo, function(result) {

```

```

    let listaArchivo = [];
    listaArchivo = result.split("\n"); //Cargamos el elemento a un Array
    y lo dividimos por linea

    //Eliminamos espacios en blanco del final
    for(let elemento=0; elemento <= listaArchivo.length; elemento++){
        if (listaArchivo[elemento]== "") {
            listaArchivo.splice(elemento - 1,2);
        }
    }

    //verificar Sintaxis
    let listaAComprobar = [];
    for (let linea = 0; linea < listaArchivo.length; linea++) {
        listaAComprobar.push(listaArchivo[linea].split(" "));
    }

    let listaCorrecta = verificarSintaxis(listaAComprobar);

    if (listaCorrecta) {

        //Ocupar los campos de la memoria para las instrucciones
        var posicionLisDir = memoriaDirection.length;
        for (let instruccion = 0; instruccion < listaArchivo.length; instruc
cion++) {
            memoriaDirection.push(zeroFill((Number(posicionLisDir) + instrucc
ion), 4));
            memoriaContenido.push(listaArchivo[instruccion]);
        }

        var memoriaDirection2 = [];
        var memoriaContenido2 = [];

        for (var i = 0; i < memoriaDirection.length; i++) {
            memoriaDirection2.push(memoriaDirection[i]);
            memoriaContenido2.push(memoriaContenido[i]);
        }

        for (let campos = memoriaContenido.length; campos <= (Number(tamMemo
ria.value) - 1); campos++) {
            memoriaDirection2.push(zeroFill((memoriaDirection2.length),4));
            memoriaContenido2.push("-");
        }

        var memoriaDirectionMostrar = [];

```

```

var posInicial = Number(kernel.value);

for (let c = (posInicial + 1); c < memoriaContenido.length; c++) {
    memoriaDirectionMostrar.push(memoriaDirection[c]);
}

//Verificar que no se exceda el tamaño de la memoria
if (((Number(listaArchivo.length) + Number(kernel.value)) > (Number(
tamMemoria.value) - 1)) || (memoriaDirection.length >= Number(tamMemoria.val
ue))) {
    alert("Espacio de memoria excedido, insuficiente espacio para las
instrucciones");
    apagar();
}

//Imprimimos el contenido de la memoria
document.getElementById("positionOnMemory").innerHTML = memoriaDirec
tionMostrar.join("<br>")
document.getElementById('operations').innerHTML = listaArchivo.join(
'<br>');
document.getElementById("directionSection").innerHTML = memoriaDirec
tion2.join("<br>")
document.getElementById("contentSection").innerHTML = memoriaConteni
do2.join("<br>");

//Imprimir Variables y Etiquetas
listaVariables = [];
listaEtiquetas = [];
for (let linea = 0; linea < listaAComprobar.length; linea++) {
    if ((listaAComprobar[linea][0]) == "nueva") {
        listaVariables.push(listaArchivo[linea]);
    } else if ((listaAComprobar[linea][0]) == "etiqueta") {
        listaEtiquetas.push(listaArchivo[linea]);
    }
}

document.getElementById("contentVariables").innerHTML = listaVariabl
es.join("<br>");
document.getElementById("contentLabels").innerHTML = listaEtiquetas.
join("<br>");

} else {

    alert("Error de sintaxis!");
    location.reload();
}

```

```

    }

    }
  )
}
}

function procesarArchivo(ch, callback) {
  var reader = new FileReader();
  reader.readAsText(ch);
  reader.onload = function () {
    callback(reader.result);
  }
}

function verificarSintaxis(lista) {

  for(let instruccion=0; instruccion < lista.length; instruccion++) {
    if(lista[instruccion][0]=='nueva') {
      return true
    }else if(lista[instruccion][0] == 'lea') {
      return true
    }else if(lista[instruccion][0] == 'cargue') {
      return true
    }else if(lista[instruccion][0] == 'almacene') {
      return true
    }else if(lista[instruccion][0] == 'vaya') {
      return true
    }else if(lista[instruccion][0] == 'vayasi') {
      return true
    }else if(lista[instruccion][0] == 'etiqueta') {
      return true
    }else if(lista[instruccion][0] == 'sume') {
      return true
    }else if(lista[instruccion][0] == 'reste') {
      return true
    }else if(lista[instruccion][0] == 'multiplique') {
      return true
    }else if(lista[instruccion][0] == 'divida') {
      return true
    }else if(lista[instruccion][0] == 'potencia') {
      return true
    }else if(lista[instruccion][0] == 'modulo') {
      return true
    }
  }
}

```

```

        }else if(lista[instruccion][0] == 'concatene') {
            return true
        }else if(lista[instruccion][0] == 'elimine') {
            return true
        }else if(lista[instruccion][0] == 'Y') {
            return true
        }else if(lista[instruccion][0] == 'O') {
            return true
        }else if(lista[instruccion][0] == 'NO') {
            return true
        }else if(lista[instruccion][0] == 'muestre') {
            return true
        }else if(lista[instruccion][0] == 'imprima') {
            return true
        }else if(lista[instruccion][0] == 'retorne') {
            return true
        }else if(lista[instruccion][0] == 'porcentaje') {
            return true
        } else {
            return false;
        }
    }
}

function zeroFill( number, width )
{
    width -= number.toString().length;
    if ( width > 0 )
    {
        return new Array( width + ( /\./.test( number ) ? 2 : 1 ) ).join( '0' ) +
number;
    }
    return number + ""; // siempre devuelve tipo cadena
}

function mostrarImpresora() {
    overlay.style.display = "block";
    cerrarVentImprimir.style.display = "block";
    cerrarVentImprimir.disabled = false;
    printSection.style.display = "block";
    printSection.style.transform = "translateY(0)";
}

function cerrarVentanaImprimir() {
    overlay.style.display = "none";

```

```
cerrarVentImprimir.style.display = "none";  
cerrarVentImprimir.disabled = true;  
printSection.style.display = "none";  
}
```