অপারেটিং সিস্টেমের ভূমিকা

(Introduction to Operating Systems)

প্রফেসর চেস্টার রেবিয়ার

(Prof. Chester Rebeiro)

কম্পিউটার সায়েন্স এন্ড ইঞ্জিনিয়ারিং

(Department of Computer Science and Engineering)

ইন্ডিয়ান ইনস্টিটিউট অফ টেকনোলজি (Indian Institute of Technology),মাদ্রাজ

সপ্তাহ – 01 বক্তৃতা- 08

সেগমেন্টেশন

নমস্কার।পূর্ববর্তী ভিডিওতে আমরা স্মৃতি ব্যবস্থাপনা বা মেমোরি ম্যানেজমেন্ট সম্পর্কে দেখেছি, বিশেষ করে আমরা ভার্চুয়াল মেমোরি নামে পরিচিত একটি ধারণার উপর দৃষ্টি নিবদ্ধ করেছি। এই ভিডিওতে আমরা মেমরি ম্যানেজমেন্টের জন্য আরেকটি গুরুত্বপূর্ণ ধারণা দেখব যা সেগমেন্টেশন নামে পরিচিত।

(স্লাইডটাইপ পড়ুন: 00:33)

সুতরাং, আমাদের প্রোগ্রামগুলিকে সাধারণত লজিক্যাল মডিউলে বিভক্ত করা যায়। সুতরাং, লজিক্যাল মডিউল যেমন উদাহরণস্বরূপ গ্লোবাল তথ্য, স্ট্যাক, হীপ, ফাংশন, ক্লাস, নামস্থান বা নেম স্পেস ইত্যাদি।

(স্লাইডটাইপ পড়ুন: 00:47)

আবার অন্য দিকে ভার্চুয়াল মেমরি প্রোগ্রামগুলিকে লজিক্যাল মডিউলে বিভক্ত করে না বরং পরিবর্তে ভার্চুয়াল মেমরিটি প্রোগ্রামগুলিকে নির্দিষ্ট আকারের ব্লকগুলিতে ভাগ করে দেয়। সুতরাং, এই যখন এটা সাধারণভাবে কাজ করতে পারে এটি করা খুব একটা লজিক্যাল হবে না। উদাহরণস্বরূপ, আমাদের একটি লজিক্যাল ব্লকে একটি ফাংশনের কয়েকটি ইন্সট্রাকসন থাকতে পারে এবং সেই ফাংশনের বাকি ইন্সট্রাকসনগুলি একটি সম্পূর্ণ ভিন্ন লজিকাল ব্লকের মধ্যে থাকে।

(স্লাইডটাইপ পড়ুন: 01:17)

অন্যদিকে সেগমেন্টেশন, প্রোগ্রামের আরও লজিকাল বিভাজন অর্জন করে। সুতরাং, আমরা কয়েকটি বাইট থেকে 4 গিগা বাইট পর্যন্ত আকারে পরিবর্তন করতে সেগুলিকে সংজ্ঞায়িত করতে পারি এবং আমরা সেগুলিকে আরও লজিক্যাল ক্রম হিসাবে নির্ধারণ করতে পারি। উদাহরণস্বরূপ, এই বিশেষ স্লাইডটি দেখানো হচ্ছে আমাদের প্রোগ্রামের মধ্যে প্রতিটি ফাংশন একটি পৃথক সেগমেন্টে থাকতে পারে।

(স্লাইডটাইপ পড়ুন: 01:39)

সেগমেন্টেশন এর একটি খুব সাধারণ ব্যবহারটি হল বিভিন্ন বিভাগ যেমন টেক্সট সেগমেন্ট, ডাটা সেগমেন্ট, হীপ সেগমেন্ট এবং স্ট্যাক সেগমেন্টে বিভক্ত করা হয়। সুতরাং, এটিই প্রোগ্রামের লজিক্যাল ভিউ হিসাবে পরিচিত হয়।

(স্লাইডটাইপ পড়ুন: 01:55)

এখন, দেখি কিভাবে অ্যাড্রেস ম্যাপিং লজিক্যাল ভিউ থেকে সেগমেন্টেশন-এ ফিজিক্যাল ভিউয়ের জন্য করা হয়। সুতরাং, মূলত আমাদের একটি সেগমেন্ট বর্ণনাকারী টেবিল আছে যা মেমরি মধ্যে সংরক্ষিত হয়। সেগমেন্ট ডেসক্রিপটর টেবিলের প্রতিটি সারি একটি নির্দিষ্ট সেগমেন্টের সাথে সম্পর্কিত। উদাহরণস্বরূপ, ডেটা সেগমেন্ট 2 একটি অফসেট 2 এ সেগমেন্ট ডেসক্রিপটর টেবিলে থাকে এবং অফসেটটি র‍্যামের বেস অ্যাড্রেস এবং সেগমেন্টের সীমা উল্লেখ করে। এখন ম্যাপিং করার জন্য, প্রসেসরের কমপক্ষে 3 টি রেজিস্টার থাকবে যা সেগমেন্ট নির্বাচক, অফসেট রেজিস্টার এবং ডেসক্রিপটর টেবিলের পয়েন্টার। সুতরাং, যেহেতু নামটি ডেসক্রিপটর টেবিলের পয়েন্টার সুপারিশ করে মেমরি অবস্থানের একটি পয়েন্টারকে যা ডেসক্রিপটর টেবিলকে পয়েন্ট করে। অন্য দিকে সেগমেন্ট নির্বাচক, ডেসক্রিপটর টেবিলের একটি অফসেট হয়। প্রসেসরের মেমরি ম্যানেজমেন্ট ইউনিট এই বিশেষ অফসেটটি সন্ধান করবে এবং বেস ভ্যালু 3000 কে বেছে নেয়। সুতরাং, এই বেস মানটি অফসেট রেজিষ্ট্রারে অন্তর্ভুক্ত করা হয় যা এফেক্টিভ ঠিকানা হিসাবে পরিচিত হয়। সুতরাং, এই এফেক্টিভ ঠিকানাটি র‍্যামের কিছু ঠিকানার সাথে মিলিত হবে।

(স্লাইডটাইম পড়ুন: 03:16)

আসুন আমরা এই ম্যাপিং স্কিমের অন্য একটি দৃষ্টিভঙ্গির দিকে তাকাই। লজিক্যাল ঠিকানা দুটি অংশ নিয়ে গঠিত, এটি একটি সেগমেন্ট নির্বাচক এবং সেইসাথে একটি অফসেট ঠিকানা যা কার্যকর ঠিকানা হিসাবে পরিচিত। সুতরাং, একটি ইন্টেল 32 বিট প্রসেসরের মধ্যে সেগমেন্ট নির্বাচক 16 বিট, যখন এফেক্টিভ ঠিকানা বা অফসেট রেজিস্টার 32 বিট এর। সুতরাং, এটা ঘটবে যে সেগমেন্ট নির্বাচকের উপাদান বিষয়বস্তু ডেসক্রিপটর টেবিলের

অফসেট হিসাবে ব্যবহার করা হয়। তারপর মেমরি ম্যানেজমেন্ট ইউনিট এই বিশেষ অফসেট থেকে বেস অ্যাড্রেস বাছাই করে এবং এফেক্টিভ ঠিকানার সঙ্গে এত জুড়ে দেয় যা লিনিয়ার ঠিকানা হিসাবে পরিচিত।

(স্লাইডটাইপ পড়ুন: 03:58)

আসুন আমরা একটি উদাহরণ সহ সেগমেন্টেশনে করা ম্যাপিংটি দেখি। আসুন আমরা বলতে পারি যে ডেসক্রিপটর টেবিলের পয়েন্টার ধারণকারী রেজিস্টারটিতে 3000 এর মান রয়েছে। তাই, এর মানে হল যে র‍্যামের 3,000 ঠিকানাতে এই ডেসক্রিপটর টেবিল রয়েছে যাতে বিভিন্ন সেগমেন্টের জন্য ম্যাপিং করা রয়েছে। এখন আসুন আমরা বলি যে সেগমেন্ট রেজিস্টারে একটি মান আছে 1, তাই এর অর্থ হচ্ছে আমরা ডেসক্রিপটর টেবিলের অফসেট 1 এই সেগমেন্ট ব্যবহার করার চেষ্টা করছি। মেমরি ম্যানেজমেন্ট ইউনিট তারপর এই অফসেটের সাথে সংশ্লিষ্ট বেস অ্যাড্রেসটি গ্রহণ করবে, এই ক্ষেত্রে 1000 ধরে নিই এবং অফসেট রেজিষ্টারটি ব্যবহার করে যার মান 100 যা 1100 এর লিনিয়ার অ্যাড্রেস হিসাবে পরিচিত হয়। এখন, অফসেট রেজিস্টার ফর্ম সহ সেগমেন্ট রেজিস্টার লজিক্যাল ঠিকানা হিসাবে পরিচিত।

(স্লাইডসময় পড়ুন: 04:51)

সুতরাং, সেগমেন্টেশনের বৃহত্তম সমস্যা হল ফ্র্যাগমেন্টেশন। আসুন এই বিশেষ উদাহরণটি দেখি; আমাদের কাছে র‍্যামে 70 কিলোবাইটের একটি ফাকা জায়গা আছে। তবে, ফাকা মেমরি সংলগ্ন অবস্থানে নেই; আমাদের একটি অংশে 60 কিলোবাইটের মুক্ত স্থান আছে, আরেকটি 10 কিলোবাইট ফ্রি স্পেস। সুতরাং, এটি 65 কিলোবাইটের একটি নতুন সেগমেন্ট বরাদ্দ করতে ব্যবহার করা যাবে না। সুতরাং, যদিও 70 কিলোবাইট ফ্রি মেমরি উপলব্ধ আছে, কিন্তু মেমরি সংলগ্ন অবস্থানে নেই এবং তাই, ব্যবহার করা যাবে না। ফ্র্যাগমেন্টেশন সেগমেন্টের সবচেয়ে বড় সীমাবদ্ধতা ; তবে ভার্চুয়াল মেমরির সাপেক্ষে ফ্র্যাগমেন্টেশনটি অনেক কম সমস্যা।

(স্লাইডটাইপ পড়ুন: 05:41)

আমরা এখন দেখব কিভাবে ইন্টেল x86 সিস্টেম সেগমেন্টেশন এবং পেজিং উভয়ই ব্যবহার করে যাতে সবরকম সুবিধাই পাওয়া যায়। সুতরাং, x86 সিস্টেমের মধ্যে, CPU দ্বারা একটি লজিক্যাল অ্যাড্রেস তৈরি করা হয় যা একটি সেগমেন্ট এবং একটি অফসেটের অংশ। সুতরাং এটি একটি সেগমেন্টেশন ইউনিট যা পাঠানো হয় আর তারপর লিনিয়র ঠিকানা তৈরি করা হয়। পেজিং ইউনিট মূলত ভার্চুয়াল মেমরি ম্যানেজমেন্ট যা লিনিয়র ঠিকানাটি গ্রহণ করবে এবং ফিজিক্যাল ঠিকানা তৈরি করবে। সুতরাং, আমরা দেখবো x86 সিস্টেমগুলিতে সেগমেন্টেশন ইউনিট কীভাবে ডিজাইন করা হয়।

(স্লাইডটাইপ পড়ুন: 06:16)

x86 সিস্টেমের 2 ধরনের ডেসক্রিপটর টেবিল আছে। একটি, স্থানীয় ডেসক্রিপটর টেবিল হিসাবে পরিচিত হয়, অন্যটি গ্লোবাল ডিসপ্লোটার টেবিল নামে পরিচিত, যা আমরা এখানে উপস্থাপন করবো। তাই গ্লোবাল ডেসক্রিপটর টেবিল মেমরিতে সংরক্ষণ করা হয় এবং এখানে এর একটি ফরম্যাট দেখানো হয়েছে। সুতরাং, মূলত এটি প্রথম ক্ষেত্র যা 0, তারপর সেগমেন্ট ডেসক্রিপটর দ্বারা অনুসরণ করা আছে। এই বিশিষ্ট গ্লোবাল ডেসক্রিপটর টেবিলটি গ্লোবাল ডেসক্রিপটর টেবিল রেজিস্টার বা GDTR নামে পরিচিত একটি রেজিস্টার দ্বারা নির্দেশিত। GDTR একটি 48 বিট রেজিস্টার, যার নিম্নলিখিত বিন্যাসটি বা ফরম্যাটটি আছে। লীস্ট সিগ্নিফিক্যান্ট 16 টি বিটের মধ্যে GDT এর আকার রয়েছে, যখন উপরের বিটগুলি বেস ঠিকানা বা GDT- এর পয়েন্টার ধরে থাকে। সুতরাং, আসুন দেখি সেগমেন্ট ডেসক্রিপটরের বিষয়বস্তু কি কি?

(স্লাইডটাইপ পড়ুন: 07:06)

সুতরাং, সেগমেন্টের ডেসক্রিপটরের 3 অংশ রয়েছে যার একটি বেস ঠিকানা আছে, এর একটি সীমা আছে এবং এর অ্যাক্সেস রাইটস আছে। বেস অ্যাড্রেস এবং সীমা 0 থেকে 4 গিগা বাইটের মান গ্রহণ করতে পারে, যখন অ্যাক্সেসের অধিকারগুলি বিট যা নির্দিষ্ট সেগমেন্টের জন্য বিভিন্ন অ্যাক্সেস নীতিগুলি রয়েছে যেমন এক্সিকিউট (Execute), রীড (Read),রাইট (Write) বা প্রিভিলেজ লেবেল (Privilege Level)।

(স্লাইডটাইপ পড়ুন: 07:30)

এবার, আসুন আমরা দেখি ইন্টেল 32 বিট মেশিনের সেগমেন্ট এবং অফসেট রেজিস্টার। সেগমেন্ট নির্বাচক রেজিস্টারগুলো 16 বিটের নির্বাচক সেগমেন্ট যা জিডিটি এর মধ্যে অফসেটকে পয়েন্ট করে। অফসেট রেজিস্টারগুলি 32 বিট রেজিস্টার হয়। সুতরাং, প্রায়শই সেগমেন্ট নির্বাচক, একজোড়া বিট যা অফসেট রেজিস্টারের অনুরূপ। উদাহরণস্বরূপ, কোড সেগমেন্ট অ্যাক্সেস করার জন্য আমরা কোড সেগমেন্টের সেগমেন্ট নির্বাচকের CS রেজিস্টার এবং সংশ্লিষ্ট ইআইপি রেজিস্টার যা অফসেট রেজিস্টার যা ইন্সট্রাকসন পয়েন্টার কে ব্যবহার করি।ডেটা সেগমেন্ট অ্যাক্সেস করার জন্য আমাদের ডিএস, ইএসএস, এফএস এবং জিএস এর মতো বেশ কিছু সেগমেন্ট রেজিস্ট্রিও রয়েছে। স্ট্যাক সেগমেন্টে অ্যাক্সেস করার জন্য আমাদের এসএস রেজিস্টার আছে যা সেগমেন্টের নির্বাচকটিকে ধরে রাখে, এবং এসপি রেজিষ্টার যা স্ট্যাক পয়েন্টারটি রাখে। জিডিটিআর এবং জিডিটি টেবিলের সাথে এই সমস্ত সেগমেন্ট নির্বাচনকারী রেজিষ্ট্রার এবং অফসেট রেজিষ্ট্রার মেমরিতে উপস্থাপন করা হয় লজিক্যাল অ্যাড্রেসকে সংশ্লিষ্ট লিনিয়ার ঠিকানাতে রূপান্তর করতে। পরবর্তী আমরা পেজিং ইউনিটটি দেখব যা মূলত x86 সিস্টেমে ভার্চুয়াল মেমরি ম্যাপিং পরিচালনা করে। সুতরাং, পেজিং ইউনিট একটি লিনিয়ার ঠিকানা নেয় এবং সমতুল্য ফিজিক্যাল ঠিকানায় রূপান্তরিত করে যা ফিজিক্যাল মেমরি অথবা RAM এর জন্য ব্যবহার করা হয়।

(স্লাইডটাইপ পড়ুন: 08:58)

x86 সিস্টেমের পেজিং ইউনিটটি 2 স্তর পেজের অনুবাদ। এটি একটি 32 বিট লিনিয়ার ঠিকানা যা 3 টি অংশে ভাগ করা হয়, সবচেয়ে গুরুত্বপূর্ণ দশটি বিট ডিরেক্টরি এন্ট্রি হিসেবে পরিচিত, এবং পরবর্তীতে টেবিলের সূচকের জন্য 10 বিট থাকে এবং অবশেষে কমপক্ষে 12 টি বিট অফসেট হয়। সুতরাং ডিরেক্টরি এন্ট্রি পেজ ডিরেক্টরির একটি নির্দিষ্ট অফসেটকে নির্দেশ করে। পেজ ডিরেক্টরি হল একটি বিশেষ টেবিল যা র‍্যামের মধ্যে উপস্থিত থাকে এবং এটি CR3 রেজিস্টার দ্বারা নির্দেশিত হয়। সুতরাং একটি পেজ টেবিলের বিষয়বস্তু নির্দিষ্ট পেজ ডিরেক্টরিকে পয়েন্ট করে এবং সেই পেজ টেবিলের মধ্যের একটি অফসেট টেবিল ইন্ডেক্স থেকে নেওয়া হয়। তাই এই নির্দিষ্ট পেজ টেবিলের বিষয়বস্তু অফসেটের সাথে তারপর ফিজিক্যাল ঠিকানা তৈরি করতে ব্যবহৃত হয়। তাই এখন আপনার জন্য দুটি প্রশ্ন আছে। এক, কতগুলি পেজ টেবিল উপস্থিত রয়েছে? সুতরাং 32 বিট ইন্টেল সিস্টেমে এই ধরনের কতগুলি টেবিল উপস্থিত রয়েছে? দ্বিতীয় প্রশ্ন হচ্ছে প্রসেসের অ্যাড্রেস স্পেসের সর্বোচ্চ মাপ কি? সুতরাং, প্রদত্ত যে প্রতিটি প্রসেস একটি,ফিজিক্যাল ঠিকানা ম্যাপিং এর একটি লিনিয়র ঠিকানা। সুতরাং, আমি চাই আপনি আসলে প্রসেসটির সর্বাধিক প্রসেস অ্যাড্রেসের আকার কি হবে তা খুঁজে বের করুন।

(স্লাইডটাইপ পড়ুন: 10:18)

এই বিশেষ স্লাইডটি x86 সিস্টেমের সম্পূর্ণ অ্যাড্রেস অনুবাদ দেখায়। একটি সেগমেন্ট নির্বাচক এবং একটি অফসেট নিয়ে গঠিত একটি লজিক্যাল অ্যাড্রেস CPU যোগ করে। সেগমেন্ট নির্বাচক হল গ্লোবাল ডেসক্রিপটর টেবিলের একটি সূচক যা সেগমেন্ট ডেসক্রিপটর নামে পরিচিত। অফসেট সহ সেগমেন্টের ডেসক্রিপটর তখন লিনিয়ার ঠিকানা তৈরী করে, এবং এই সমগ্র স্থানটি প্রসেসটির জন্য লিনিয়র অ্যাড্রেস ম্যাপ হিসাবে পরিচিত। রৈখিক ঠিকানাটিতে 3 টি উপাদান রয়েছে যা ডাইরেক্টরি এন্ট্রি, সারণি সূচক এবং অফসেট। সুতরাং, ডিরেক্টরি ডিরেক্টরি সূচী সূচী সূচী এবং এই কন্টেন্ট তারপর আপনি একটি পৃষ্ঠার টেবিলের নির্বাচন করুন। টেবিলের সূচকটি যে পৃষ্ঠার টেবিলের মধ্যে অফসেট পেতে ব্যবহৃত হয় এবং এর বিষয়বস্তু, সঙ্গে বা লাইনের ঠিকানাতে চূড়ান্ত 12 বিটগুলি তখন চূড়ান্ত শারীরিক ঠিকানাটি প্রাপ্ত করার জন্য ব্যবহৃত হয় যা র্যামে তথ্য পড়তে বা লিখতে ব্যবহৃত হয় । ভিডিওগুলির এই সেটগুলির সাথে আমরা মেমরি ম্যানেজমেন্ট স্কিম যেমন ভার্চুয়াল মেমরি এবং সেগমেন্টেশন দেখেছি এবং আমরা দেখেছি কিভাবে 32 বিট সিস্টেমে ইন্টেল অ্যাড্রেস অ্যাড্রেস অ্যাড্রেস ব্যবহার করে। ধন্যবাদ.